RightWON MODBUS Protocol – Manual - V1.2

Document No. RWM002025-MA-en

vizimax

www.vizimax.com

Copyright

Document Disclaimer

Vizimax believes the information in this document to be accurate at the time of publication of the document. However, this document may contain errors or omissions. Vizimax makes no warranties regarding this document or its content. Vizimax is not responsible for any loss, liability, or damages of any kind arising from or related to this document or the information contained in it. Information contained herein may be periodically updated or modified without notice in subsequent editions. If you encounter an error in this document, please contact Vizimax.

Any representations or statements in this document concerning Vizimax products are for informational purposes only and do not constitute warranties, either express or implied, regarding such products. Vizimax's standard limited warranty, stated in its sales contract or order confirmation form, is the only warranty offered by Vizimax in relation to the products.

All specifications and designs are subject to change without prior notice. Vizimax reserves the right, at its sole discretion, to modify or replace any part of this document. It is your responsibility to check this document periodically for changes.

Warranty Disclaimer

Vizimax and its suppliers and licensors hereby disclaim all warranties of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose and non-infringement. Neither Vizimax nor its suppliers and licensors, makes any warranty that Vizimax products will be error-free, or that access to remote units and their connected equipment will be continuous or uninterrupted.

Limitation of Liability

Vizimax products are programmable devices which can be modified by anyone who has been given authorization to access the product and log onto its configuration software. Vizimax cannot oversee modifications to the configuration of Vizimax products unless a prior service agreement has been made between all parties involved. Vizimax does not have any control over who is given permission to access Vizimax products. Vizimax cannot therefore be held responsible for the configuration, automatisms and actions that are programmed into any Vizimax product once it has been delivered to the purchaser or third party. Likewise, Vizimax is not responsible for the particular use of Vizimax products in industrial, commercial or other applications, nor is it liable for any harmful effects of such use.

You are responsible for taking precautions as necessary to protect yourself, your computer networks and all connected equipment from any harmful or destructive actions that may arise due to incorrect programming of Vizimax products, or any purposely destructive actions programmed by a person with malicious intent. Vizimax disclaims any responsibility for any harm resulting from the use of the RightWON.

In no event will Vizimax, or its suppliers or licensors, be liable with respect to any subject matter under any contract, negligence, strict liability or other legal or equitable theory for: (i) any special, incidental or consequential damages; (ii) the cost of procurement or substitute products or services; (iii) interruption of use, or the loss or corruption of data.

Vizimax, its contractors, licensors, and their respective directors, officers, employees and agents, are hereby indemnified from any and all claims and expenses, including attorneys' fees, arising out of your use or misuse of Vizimax products. Vizimax shall have no liability for any damage, injury, failure or delay due to matters beyond the reasonable control of Vizimax.

The foregoing shall not apply to the extent prohibited by applicable law.

Trademarks

Vizimax, the Vizimax logo, RightWON, WiseWON and the RightWON icons are trademarks or registered trademarks of Vizimax, Inc. in Canada, the United States and other jurisdictions. All other trademarks, registered trademarks and service marks are the property of their respective owners.

Your use of Vizimax products grants you no right or license to reproduce or otherwise use any Vizimax or third-party trademarks.

Vizimax is a licensed user of the following service marks:

Contents

Introduction 1
1.1. Document Scope 1
1.1.1. Other documents 1
1.2. Document conventions 1
1.3. Safety precautions 1
1.3.1. Warnings ⚠ 2
1.3.2. Cautions ⚠ 2
MODBUS Protocol Management 3
2.1. Introduction to the MODBUS protocol 3
2.2. Integrating the MODBUS protocol in the RightWON 4
2.2.1. Support of MODBUS protocol variants in the RightWON 4
2.2.2. MODBUS request and variables 5
2.2.3. Data types 6
2.2.4. Limitations of the MODBUS implementation 6
2.3. Configuring the hardware and the network 6
2.3.1. Configuring an adapter in the hardware configurator 6
2.3.2. Configuring a MODBUS serial link in the network configurator 7
2.3.3. Configuring a MODBUS-over-Ethernet link in the network configurator 9
2.4. Tutorial on configuring the MODBUS Slave protocol 11
2.4.1. Managing the status bits associated with the variables 11
2.4.2. Adding the MODBUS Slave protocol to the RightWON configuration 12
2.4.3. Adding and configuring a Slave server 14
2.4.4. Inserting and configuring a data block (request) 14
2.4.5. Configuring to open a communication port 18
2.4.6. Inserting and configuring a data exchange variable 23
2.5. Tutorial on configuring the MODBUS Master 26
2.5.1. Adding the MODBUS protocol to the RightWON configuration 26
2.5.2. Inserting and configuring a communication port 27
2.5.3. Inserting and configuring a data block (request) 31
2.5.4. Inserting and configuring a variable for data, status or control exchange 34
2.5.5. Example of sending a request on call 38

Revision History

Date (yy-mm-dd)	Comments	Author
2012-01-11	V1.1: Initial release	C. Archambault, J. Warne
2012-01-12	V1.2: Updated section on configuring to open a communication port.	C. Archambault

Document Applicability

This document applies to the following RightWON Configuration Suite software versions:

Document version	Product version	Comments
V1.0	1.7.0 and higher	

Introduction

Thank you for the trust you place in us, and congratulations for choosing the RightWON system from Vizimax! For your satisfaction and the success of your projects, the RightWON system has been designed and manufactured to the highest quality and performance standards in the industry.

1.1. Document Scope

This document describes the integration of the MODBUS communication protocol in a RightWON system using the RightWON Configuration Suite.

1.1.1. Other documents

For further details on the information in this document, refer to the specific manuals below:

Reference No.	Document Name
RWM000010-MA	RightWON Configuration Suite Manual
RWM000011-MA	RightWON Configuration Suite – Installation Guide
RWM000020-MA	RightWON – User Guide
RWM000050-MA	RightWON Satellite – User Guide
RWM000060-MA	RightWON Engine IEC61850-3 – User Guide
RWM000061-MA	RightWON Engine Rackmount – User guide
RWM000062-MA	RightWON Engine Standalone – User Guide
RWM000080-MA	RightWON Configuration Suite – Application Guide

1.2. Document conventions

To facilitate the reading of this document the following conventions are used:

- Menu/dialog controls and items are in **bold**, e.g. **Options/Advanced settings...**, as are buttons, e.g. **OK**
- Names of Categories, Users, Sectors and Tags defined by the system integrators are in *italics*, e.g. *John Smith*, *Generator*
- Application-specific items such as Sector, Tag, Category and User group begin with a capital letter
- Hyperlinks are in blue
- The ⚠ symbol is used to raise the reader's attention.

1.3. Safety precautions

To ensure the safety of personnel and products, and to prevent the risk of accident, you must strictly follow the cautions and warnings written on product labels, in the manuals and on the RightWON product packaging.

To ensure proper operations of the RightWON product, read this manual in its entirety before proceeding to the other stages of learning, hardware installation, configuration or operation. Make sure that you fully understand the product and all information provided in this manual. For further information or if you require assistance from Vizimax, write to our application engineers or the Technical Support group at support@vizimax.com (certain fees and conditions may apply, depending on the type of service requested).

1.3.1. Warnings ⚠

RightWON products are not designed for safety management applications or as security devices. Mishandling of this product could cause critical situations leading to personal, equipment or property damage, network failure, loss of data, electrical shock, serious injury or even death. To prevent such events from occurring:

- Take all possible measures to ensure the security of your systems through the use of appropriate equipment that meets the requirements of the application. This will help preserve the integrity of your systems in the event of product failure or other external factors.
- To prevent the risk of explosion, do not use RightWON products in areas where explosives are stored without taking appropriate measures as defined by the standards and regulations in effect, obtained from the proper local authorities.
- To prevent damage to electronic components, do not expose this product to open flame or submit it to environmental factors that exceed the recommended levels.
- Batteries may explode if they are not handled with care. Do not recharge, disassemble or dispose of in fire. We recommend that you recycle these items by taking them to the appropriate collection service.

1.3.2. Cautions ⚠

- Make sure that RightWON products are managed by qualified personnel who have been properly trained to install, configure and troubleshoot them.
- Always configure and operate this product within the recommended technical specifications and operating criteria recommended by Vizimax, as cited in this manual and the other technical documents available.
- Use homologated external emergency devices, including but not limited to: emergency stop, emergency signaling, interlock and safety circuitry.
- Properly connect and secure removable cables and connectors. Loose connections could overheat and catch fire.
- Protect all power supplies and connect to ground on the equipment using an appropriate connection. Failure to protect and/or ground the equipment could lead to fatal electrical shock.
- Take all possible measures to prevent foreign materials from falling into the product interior (liquids, flammable materials, metal objects, etc.).
- Turn the equipment off and disconnect all sources of power before undertaking any procedure whatsoever on the equipment.

MODBUS Protocol Management

This chapter deals with managing the MODBUS communication protocol, which is used to exchange data reliably between partners on a link or over a communication network. The following topics are addressed:

1- Introduction to the MODBUS protocol: describes the MODBUS protocol.

2- Integrating the MODBUS protocol in the RightWON: describes the MODBUS variants supported by the RightWON, MODBUS requests and variables, data types, and limitations for implementing MODBUS in the RightWON.

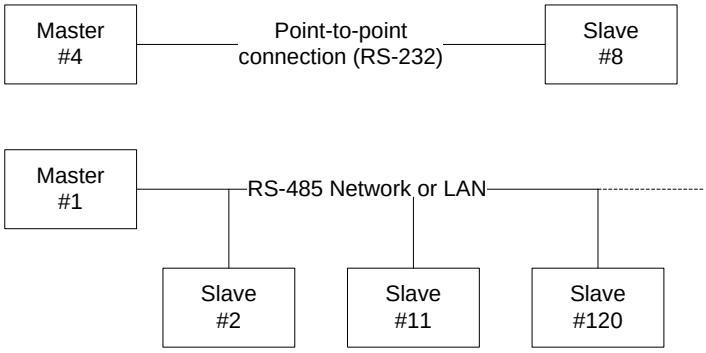
3- Configuring the hardware and the network, describes how to configure RightWON systems in the hardware and network configurators according to the MODBUS protocol variant used.

4- Tutorial on configuring a MODBUS Slave application, describes adding and configuring the MODBUS Slave protocol in the RightWON.

5- Tutorial on configuring a MODBUS Master application, describes adding and configuring the MODBUS Master protocol in the RightWON.

2.1. Introduction to the MODBUS protocol

The MODBUS protocol supports the exchange of data between a master partner and one or more slaves. Each partner has a unique 8-bit address (from 1 to 247, decimal). Address 0 is reserved for broadcast requests, and the address range from 248 to 255 is reserved for internal use.



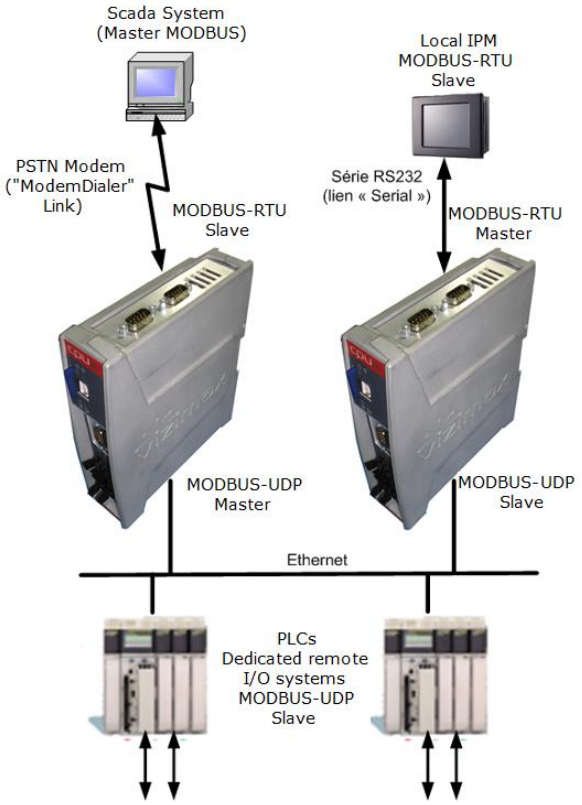
Normally, data exchange between partners is always initiated by the master through a request. This request specifies the address of the slave station, the function to be performed (request) and the address range and data, if required. This frame is validated by a checksum. When a slave receives a request it validates that it is the intended recipient, interprets the request and formulates a response. In cases where the message is not understood by the slave, a timeout management function located in the master permits recovery of the exchange. To avoid contention, the master initiates one data exchange at a time with the slaves. In addition, slaves do not transmit unsolicited data.

The MODBUS protocol supports broadcasting messages to address 0. In this mode, all slave stations receive the request, but do not formulate a response in return. Thus there is no guarantee of message delivery in this mode.

2.2. Integrating the MODBUS protocol in the RightWON

The RightWON system can be used as a server (MODBUS Slave) or client (MODBUS Master). The MODBUS server and client can be active simultaneously in a single RightWON. This allows for numerous configurations illustrated in the following figure, such as:

- Connecting a Slave to a MODBUS Master such as a supervisor (SCADA).
- Connecting a Master to MODBUS inputs/outputs.
- Connection between RightWON systems for data exchange in real time.



MODBUS protocol configuration is carried out using the RightWON Configuration Suite. This protocol is fully integrated with the application code downloaded to the RightWON system, and requires no user license.

2.2.1. Support of MODBUS protocol variants in the RightWON

RightWON systems support the MODBUS protocol on a serial link (direct or modem) or over an Ethernet network. Since message encapsulation is different for serial and Ethernet modes, the MODBUS protocol cannot be routed between a TCP port and a serial port on the RightWON. However, it can be sent via the TCP gateway to a serial TCP2Conn service. (For details, refer to the section on the TCP2Conn service in the manual "RWM000010-MA-en, RightWON Configuration Suite - Manual".)

The following variants are supported by the RightWON:

- **Serial MODBUS-RTU** standardized over a serial or modem link.
- **Open MODBUS over Ethernet TCP:** This is standard MODBUS that includes a 6-byte header (MBAP), but excludes a cyclic redundancy check (CRC16).

- **Open MODBUS over Ethernet UDP** (datagram mode): This implementation does not ensure the integrity of information transport. It uses the same frames as for the Open MODBUS over Ethernet version, but without message acknowledgment.
- **MODBUS-RTU over Ethernet UDP:** This implementation does not ensure the integrity of information transport. It uses the same frames as for the MODBUS-RTU version, but without message acknowledgment.

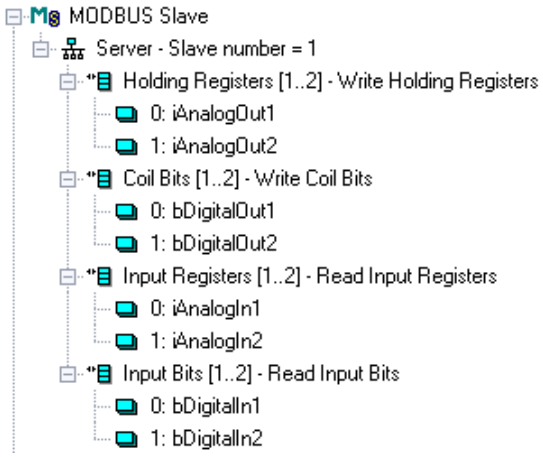
The ASCII version of the MODBUS protocol over a serial link is not directly supported, and requires creating and interpreting messages with the SERIO function. For further details, press **F1** to access the RightWON Configuration Suite help topics, and search **SERIO**.

2.2.2. MODBUS request and variables

Each request permits carrying out specific functions such as writing or reading data. Write requests are directed to the Slave, whereas read requests carry the data from the Slave to the Master. The following table lists the functions supported by the RightWON.

MODBUS Function (decimal)	Request Name	Description
1	Read coil status	Reads output bits
2	Read bit input status	Reads input bits
3	Read holding registers	Reads output words
4	Read input registers	Reads input words
5	Write single coil	Writes one output bit
6	Write single register	Writes one output word
15	Write multiple coils	Writes N output bits
16	Write multiple registers	Writes N output words

Each function conveys one or more data of the same type (Boolean, whole integer, etc.). In the RightWON, each data item corresponds to a Variable that is assigned to a specific address of a protocol function. In the following example of a MODBUS Slave implementation in the RightWON, the Slave at address 1 responds to 4 different requests. The first request involves the Master writing 2 integer (INT) output words to the RightWON (function 16). This information is conveyed to the addresses of bits 1 and 2 [1..2], which correspond to the variables *iAnalogOut1* and *iAnalogOut2*, respectively.



2.2.3. Data types

BOOL, INT and UINT variables can be associated with MODBUS data. STRING variables are not supported and cannot be exchanged using MODBUS. The 64-bit variables (LINT and LREAL) cannot be exchanged directly without loss of accuracy or value. To exchange 32-bit variables (DINT, REAL, ...), you can specify that the variable be linked to two consecutive MODBUS words.

To exchange Boolean variables in MODBUS words, you can specify a hexadecimal mask that indicates to which bit of the word the variable is linked. For example, enter the mask "0001" to connect a variable to the least significant bit of the word. For further details on masks, refer to the section on managing address discontinuities.

2.2.4. Limitations of the MODBUS implementation

The MODBUS client function (Master) opens a communication port (serial or Ethernet) for each MODBUS port configured. The number of client ports that can be open at the same time is unlimited.

On the MODBUS server side (Slave), communication management (sending and receiving frames) is fully integrated with the communication server in the RightWON.

Limitations on the Slave are:

- One serial link per Slave.
- A maximum of 7 ports on the Ethernet link.

Additional Slave ports can be opened using the MBSlaveRTU or MBSlaveUDP function blocks. For further details about these blocks, consult the help function of the RightWON Configuration Suite by pressing the **F1** key, and search **MBSlaveRTU** or **MBSlaveUDP**.

2.3. Configuring the hardware and the network

You must configure the hardware and network configurators of the RightWON systems according to one of the MODBUS protocol variants, namely:

- **MODBUS over Ethernet:**
    a. Configure an Ethernet adapter
    b. Configure an Ethernet link
- **Serial MODBUS-RTU:**
    a. Configure a serial adapter
    b. Configure a serial link

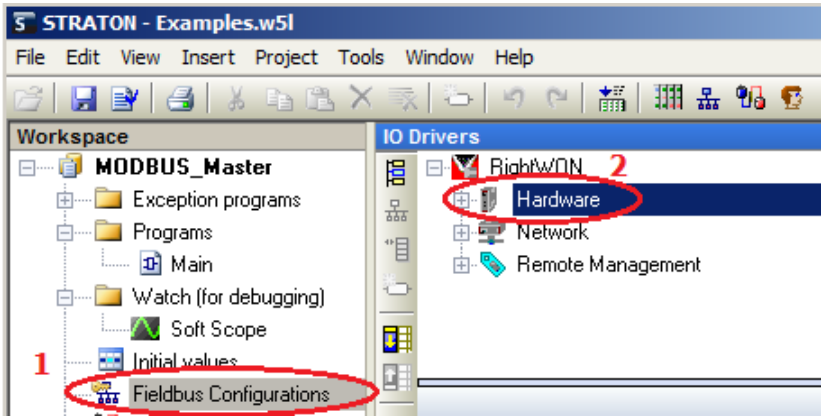
2.3.1. Configuring an adapter in the hardware configurator

For the **serial MODBUS-RTU** protocol, configure a serial adapter. For **MODBUS over Ethernet**, configure an Ethernet adapter.

To install in an expansion module (e.g. RWC 0T0000), carry out the following steps:

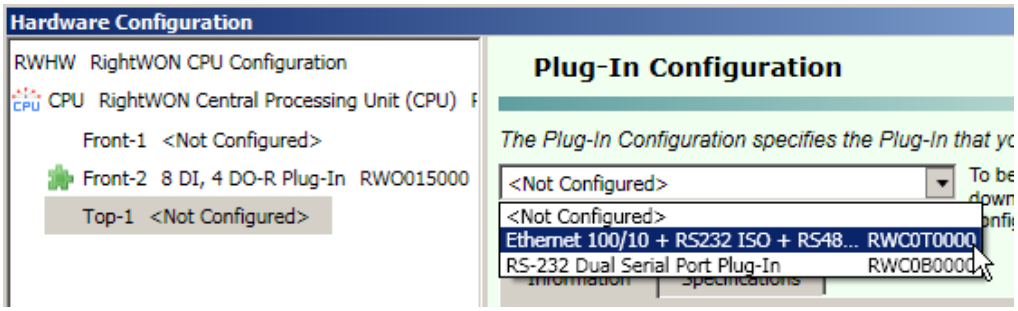
1- In the work area, click **Fieldbus Configurations**  .

2- Double click on **Hardware**  .



3- In the navigation area, select **Front-1**, **Front-2** or **Top-1**, depending on where the Plug-In is installed in the RightWON Satellite (e.g. TOP-1 for the RWC 0T0000 module). In the work area, choose the expansion module (e.g. RWC 0T0000) from the dropdown menu.

Note: For further details, refer to the section on the RightWON hardware configurator in the guide "RWM000010-MA-en, RightWON Configuration Suite - Manual".



4- Under the **Specifications** tab, verify the technical specifications of the expansion module in order to configure the hardware and adapters (**Adapters**) according to the protocol variant used. Ensure that the communication speed and other adapter fields are configured identically on both Master and Slave.

5- Click **OK**.

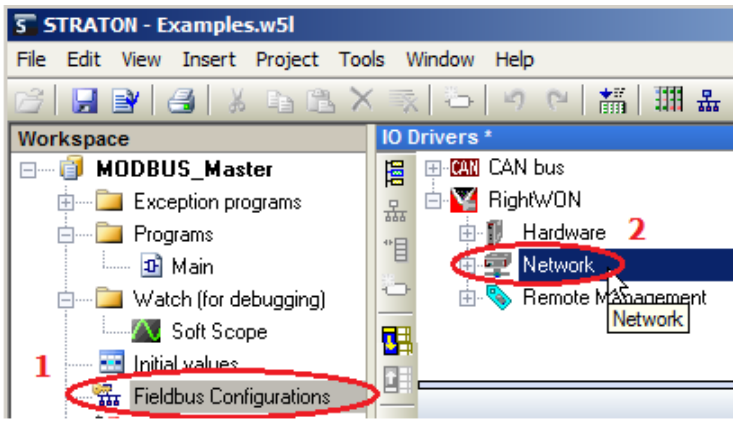
6- You can now configure a link in the network configurator according to whether you are using MODBUS over Ethernet or serial MODBUS-RTU.

2.3.2. Configuring a MODBUS serial link in the network configurator

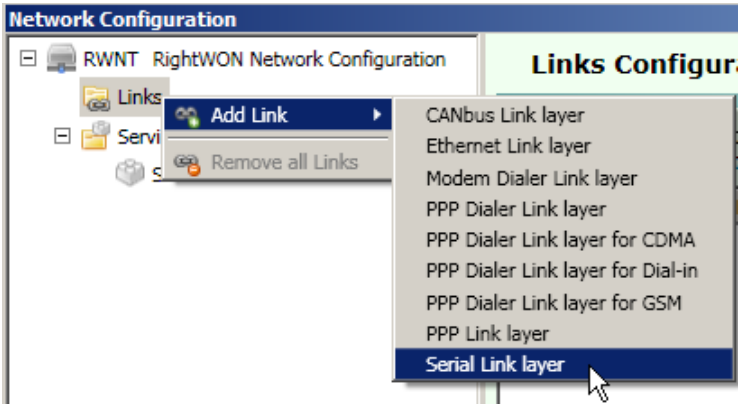
To add a serial link in the network configurator, carry out these steps:

1- In the work area, click **Fieldbus Configurations** .

2- Double click on **Network**.

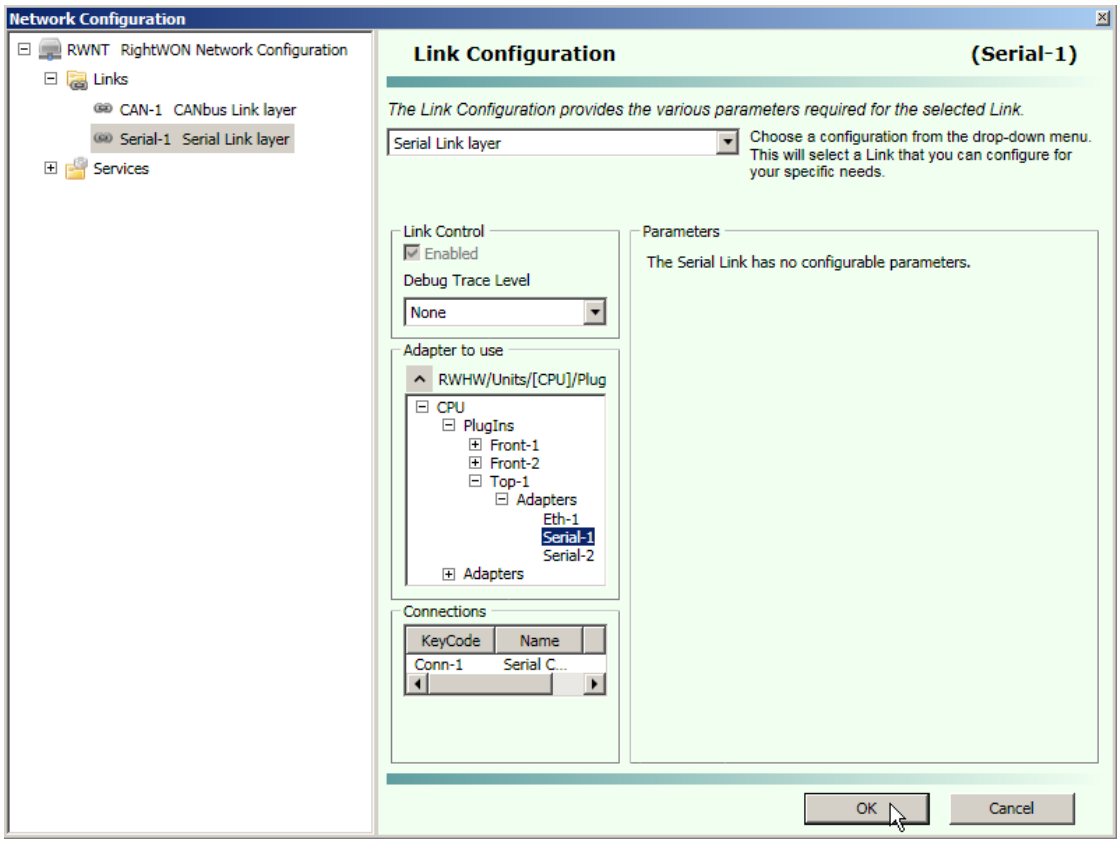


3- Open the **RWNT** tree view. Right click on **Links** and select **Serial Link Layer**.



4- In the navigation area, click on the **Serial-1** link. Under **Adapter to use**, select the **Serial-1** adapter at the location where the Plug-In is installed in the RightWON (e.g. **CPU/PlugIns/Top-1/Adapters**).

5- To finish, click **OK**.



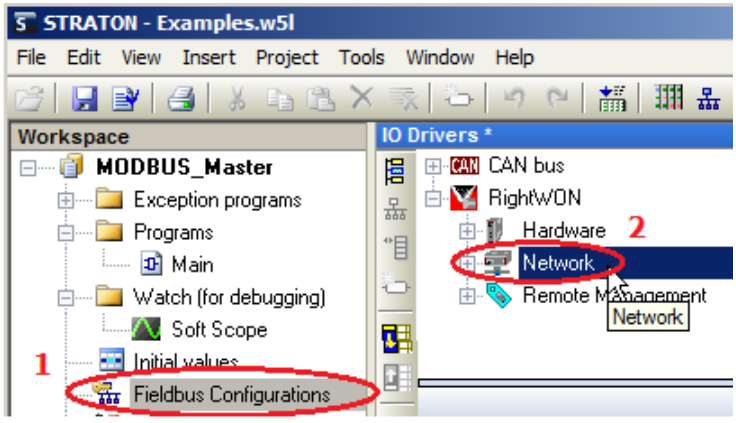
6- You can now configure the MODBUS Slave and/or MODBUS Master.

2.3.3. Configuring a MODBUS-over-Ethernet link in the network configurator

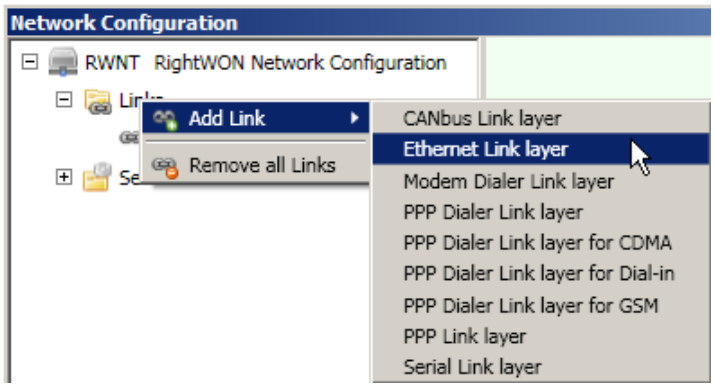
To add an Ethernet link in the network configurator, carry out the following steps:

1- In the work area, click **Fieldbus Configurations**.

2- Double click on **Network**.



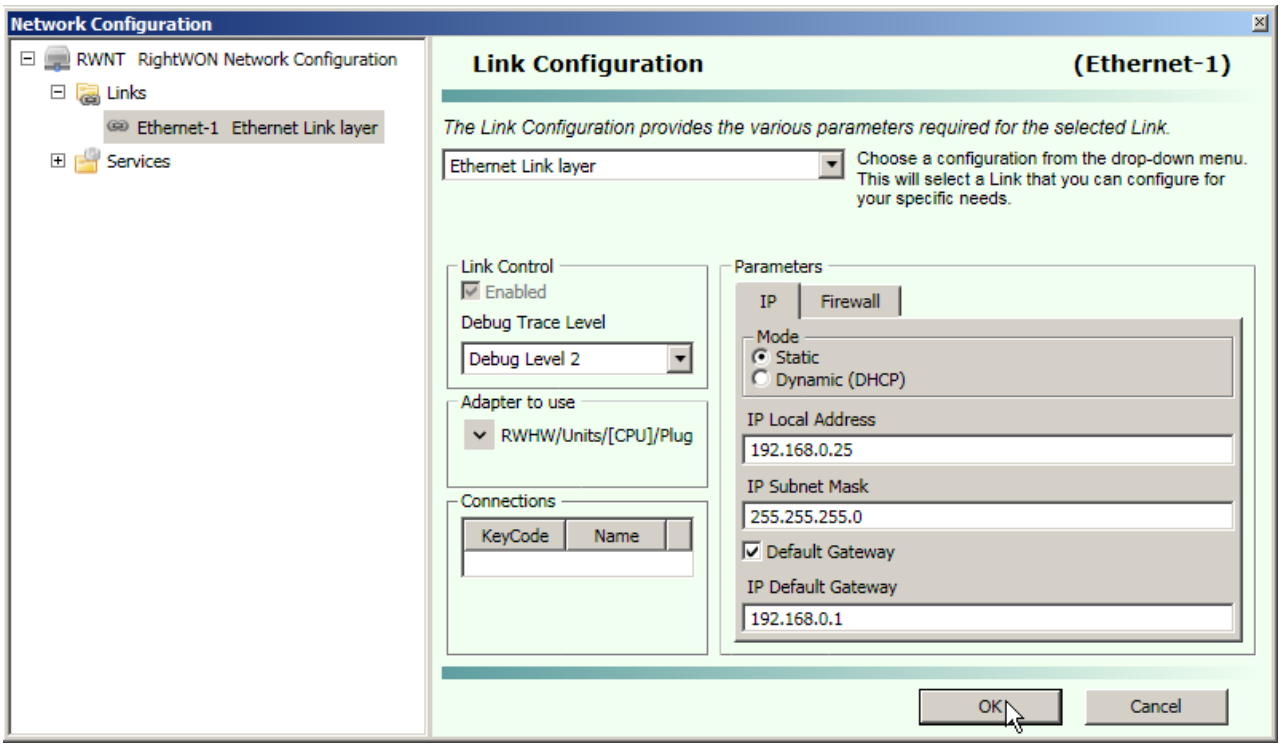
3- Open the **RWNT** tree view. Right click on **Links** and select **Ethernet Link Layer**.



4- In the navigation area, click on the **Ethernet-1** link. Under **Adapter to use**, select the **Eth-1** adapter at the location where the Plug-In is installed in the RightWON (e.g. **CPU/PlugIns/Top-1/Adapters**).

5- Under the **IP** tab, the Ethernet link is configured manually with a static IP address (**Static**) or automatically with an IP address from the DHCP server (**Dynamic DHCP**). The TCP-IP stack will route messages over the appropriate Ethernet interface using the following rules:

- If the IP address (**IP Local Address**) of the MODBUS Slave device is in the same range as that indicated by the subnet address (**IP Subnet Mask**) of the MODBUS Master device, it is assumed that the devices are on the same network segment.
- If the range defined by the **IP Subnet Mask** of the MODBUS Master device does not include the IP address (**IP Local Address**) of the MODBUS Slave device, it is assumed that the device is not on the same network segment. In this case the message will be routed to the default gateway (**IP Default Gateway**).
- If the IP address (**IP Local Address**) cannot be resolved on the Ethernet links, the TCP-IP stack will attempt to route the message over a PPPDialer link, if configured.



6- To finish, click **OK**.

7- You can now configure the MODBUS Slave and/or MODBUS Master.

2.4. Tutorial on configuring the MODBUS Slave protocol

If you have not already done so, you must configure the hardware and the network first. Then follow these steps to configure a MODBUS Slave in the RightWON:

1- Manage the status bits associated with the variables.

2- Add the MODBUS Slave protocol to the RightWON configuration.

3- Insert and configure a server for the MODBUS Slave protocol.

4- Add and configure a data block (session), to permit the insertion of variables when configuring the block.

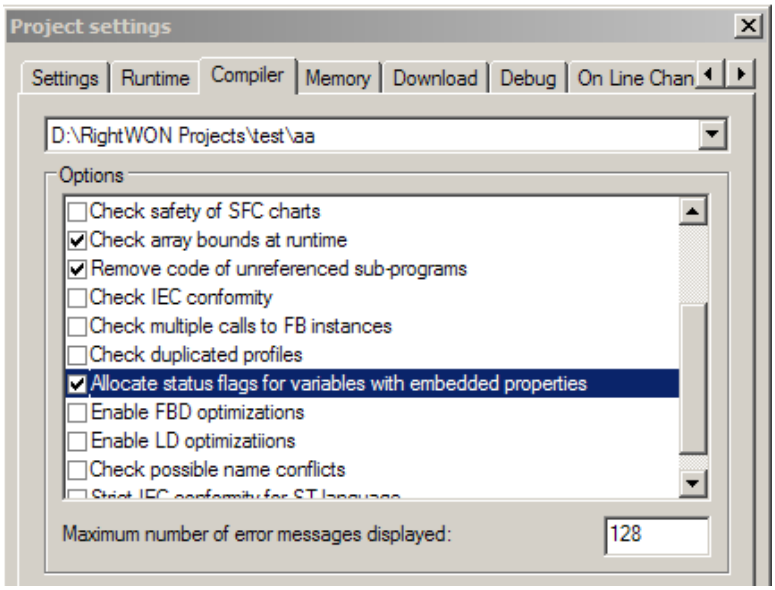
5- Open a communication port to allow the MODBUS Slave to communicate with the MODBUS Master.

2.4.1. Managing the status bits associated with the variables

In addition to updating the variables associated with the protocol, the MODBUS Master manager is able to manage a status bit associated with each variable. This status bit (*_VSB_I_BIT*) indicates whether data exchange with the partner is working properly. When a read request does not work (timeout, communication error with the partner, etc.), this bit is TRUE. Note that the status bits of variables associated with MODBUS protocol write requests are not handled automatically and must be processed via the application.

Two things are required to manage the status bit associated with each variable:

1. Choose the compiler option "**Allocate status flags for variables with embedded properties**" from the **Project/Settings/Advanced/Compiler** menu of the Configuration Suite.



2. Associate a profile with the variable. A profile is automatically associated with a variable that is paired with an energy protocol such as DNP3 or IEC61850, or with a remote management Tag.

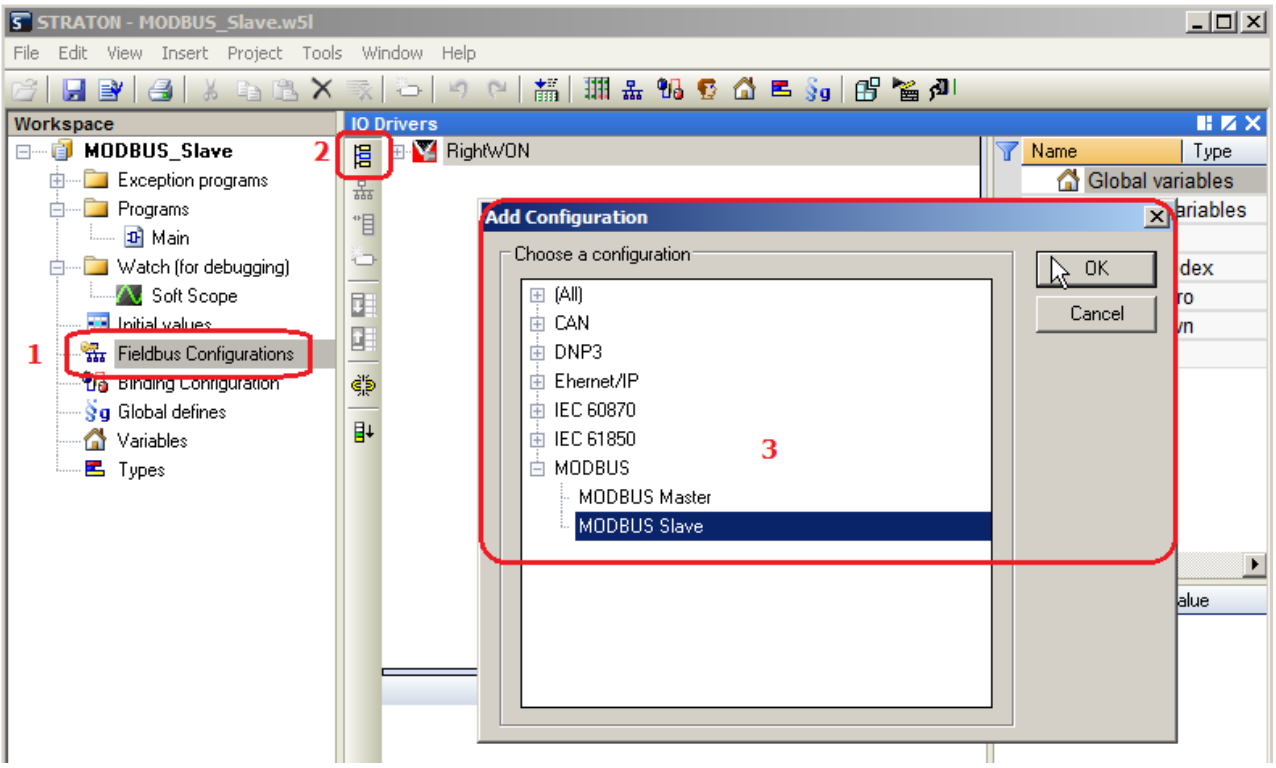
2.4.2. Adding the MODBUS Slave protocol to the RightWON configuration

The MODBUS Slave can be added using the fieldbus manager, by carrying out the following steps:

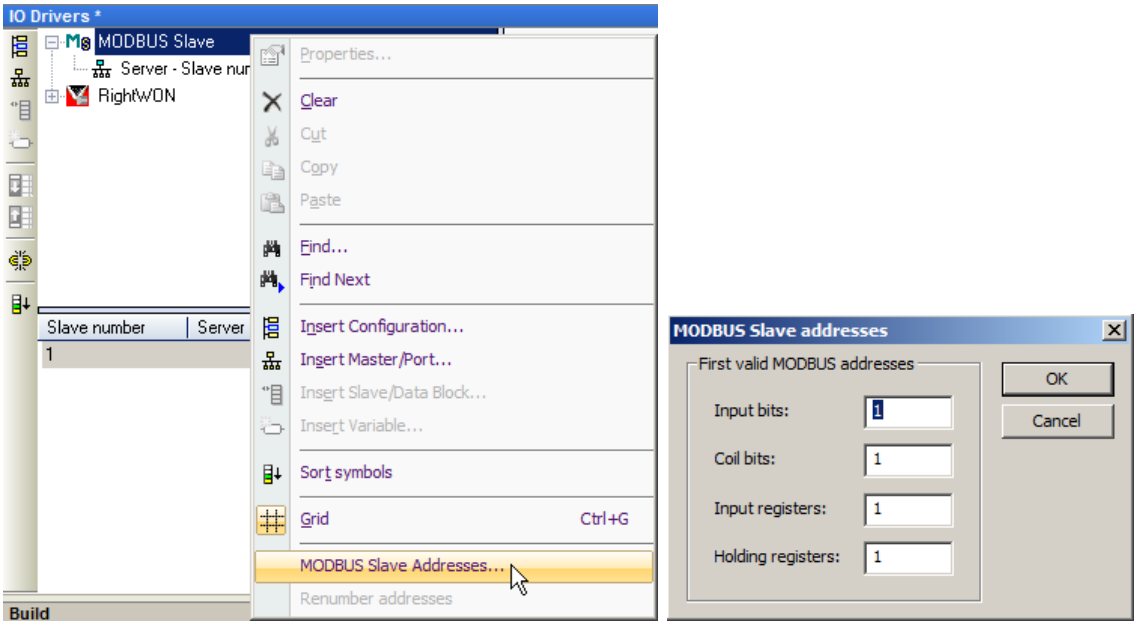
1- Click on **Fieldbus Configurations** in the work area.

2- Click on **Add configuration...** in the editing area toolbar.

3- Expand the **MODBUS** configuration and click on the **MODBUS Slave** protocol. Click **OK**.



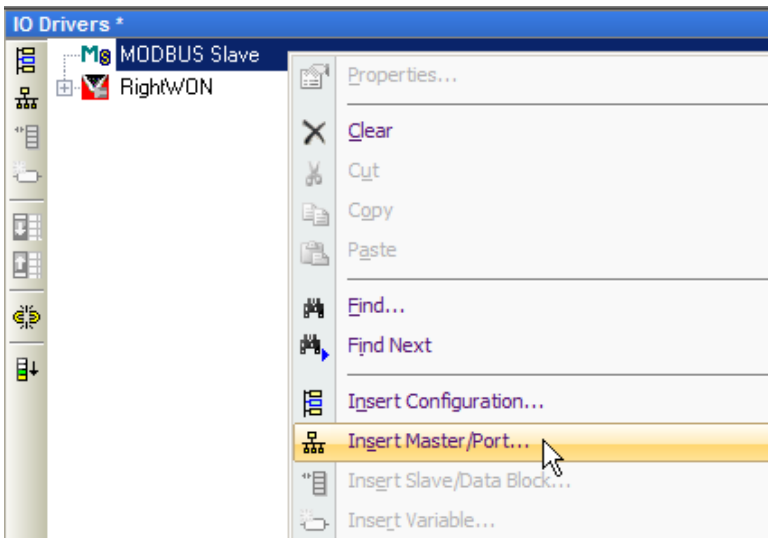
4- By default, the first valid address of each MODBUS function is 1. To use MODBUS equipment that has other addressing conventions, you must change the base offset for each data type. To do this, right click on **MODBUS Slave** and click on **MODBUS Slave Addresses...**



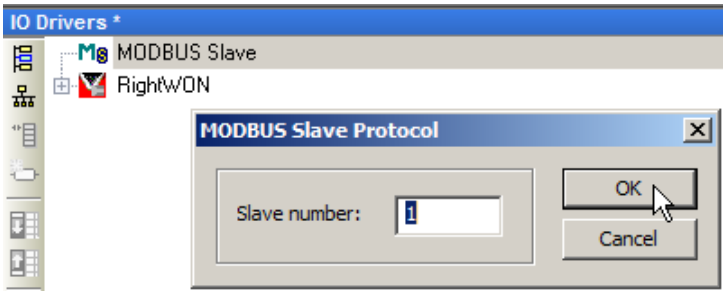
2.4.3. Adding and configuring a Slave server

Insertion of the communication port (server) for the MODBUS Slave is carried out as follows:

1- From the **IO Drivers** window, right click on **MODBUS Slave**, then click on **Insert Master/Port...**



2- In the **Slave number** field, enter a MODBUS address that is unique for the communication link on which the RightWON is installed. Addressing is in decimal and the valid range is from 1 to 247, with address 0 being reserved for broadcasting and 248 to 255 for internal use.

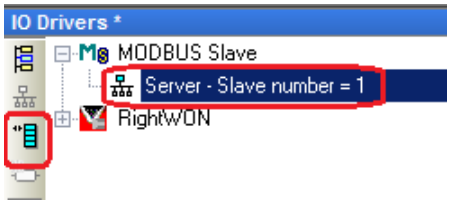


3- Click **OK** to create the new configuration.

2.4.4. Inserting and configuring a data block (request)

1- Click on **Server – Slave number** in the **IO Drivers** window.

2- Click on **Insert Slave/Data clock...** in the toolbar.



3- The **MODBUS Slave Request** window permits configuration of a frame or communication exchange:

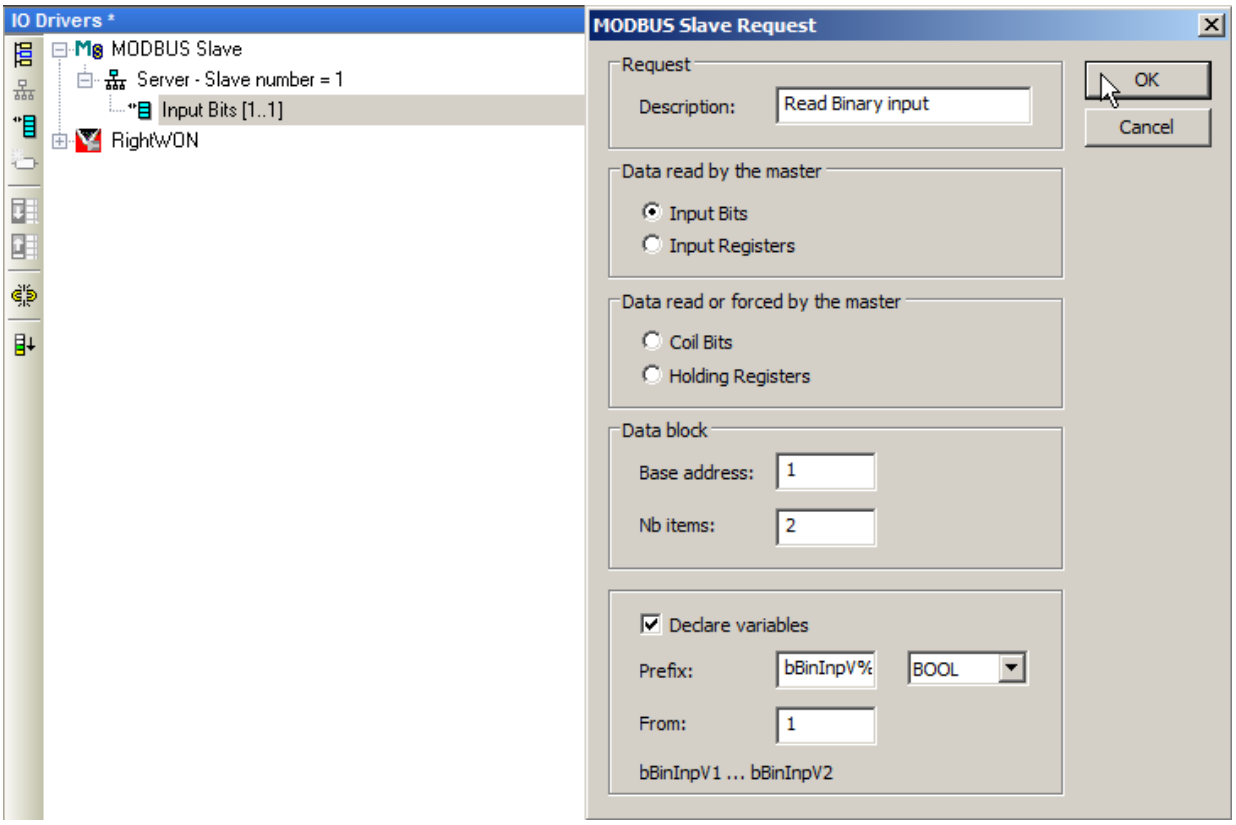
- In the **Request** section, the **Description** field permits entering a description of the request.
- The following request types are available:
  - **Input Bits:** binary values read by external Masters (function 2).
  - **Input Registers:** analog words read by external Masters (function 4).
  - **Coil Bits:** binary values forced by external Masters (function 5 or 15).
  - **Holding Registers:** analog words forced by external Masters (function 6 or 16).
- Each block is identified by a **Base address** (in decimal) and the number of variables, **Nb items** (bits or words).

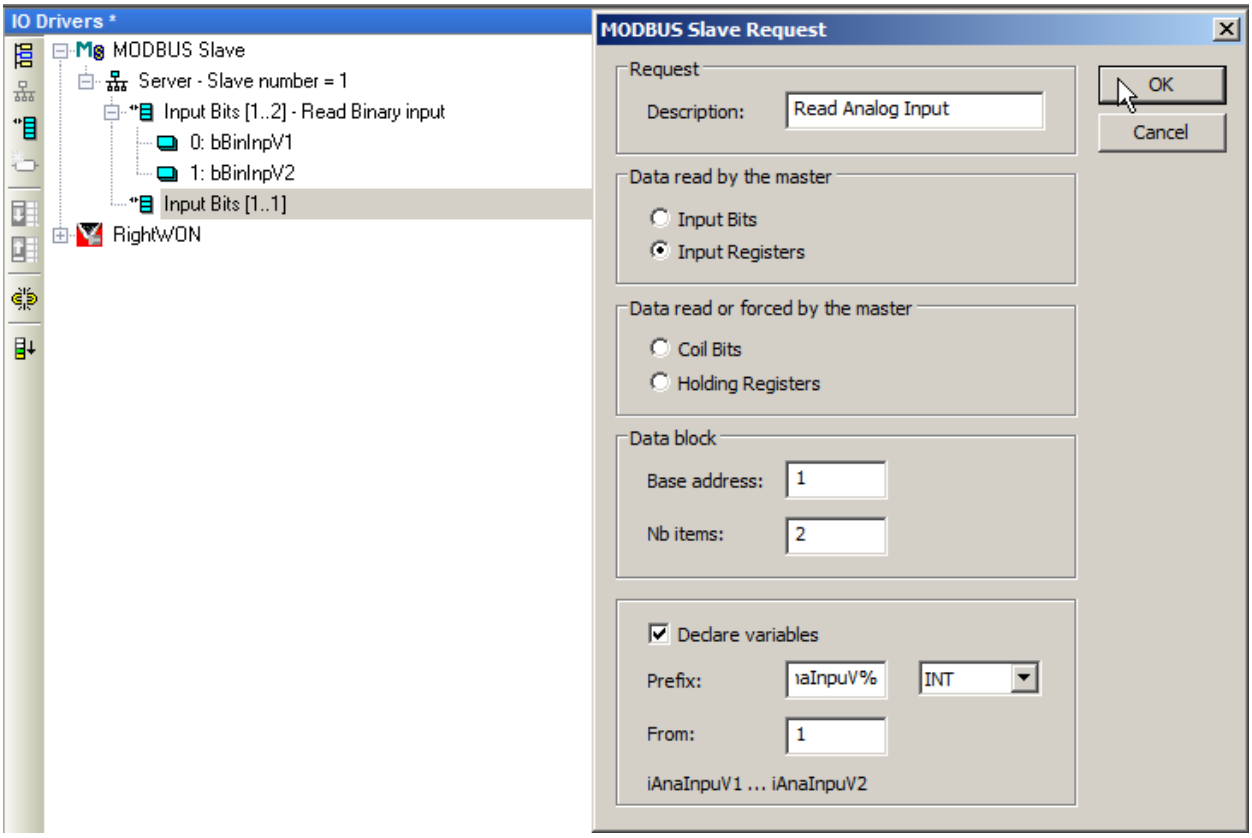
  ⚠ **Caution!** Read/write requests received from the MODBUS Master will be rejected if the address range specified in the request is not included in any of the MODBUS Slave blocks that are configured. Requests that intersect with several configured blocks will also be rejected. For example, if you configure a 16-word block at address 1 and another 16-word block at address 17, a 32-word read or write request starting from address 1 will be refused and an "Invalid address" report will be returned.
- To insert the variables in the data block, check the **Declare variables** box. Enter the prefix for the variables in the **Prefix** field followed by **%** (e.g. **V%**). For a word-based request, choose the type of variable (e.g. INT or UINT) from the dropdown menu. This will automatically create the variables in the data block and in the global variables according to the **Base address** and number of items (**Nb items**).

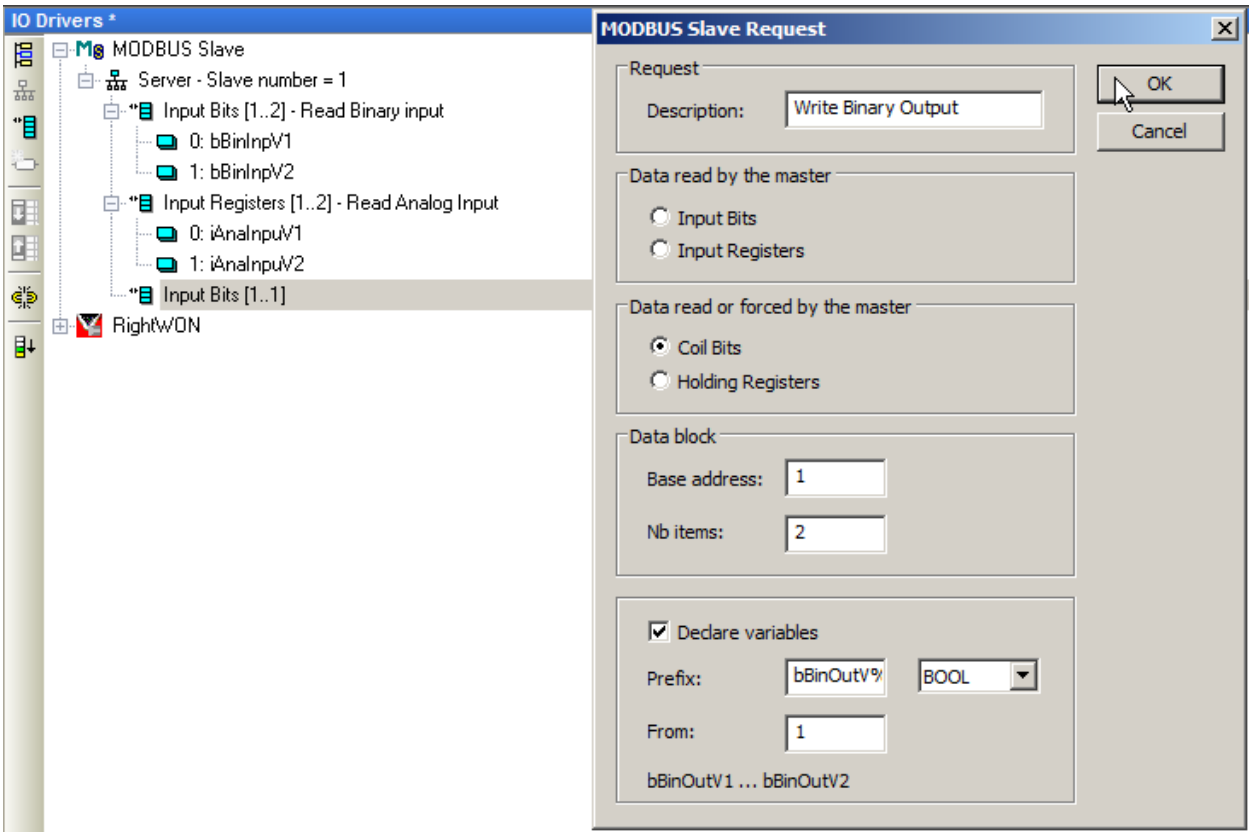
  Note: To transfer variables other than BOOL, INT and UINT or to specify a mask on a word, uncheck **Declare variables** and add the variables manually to the data block.

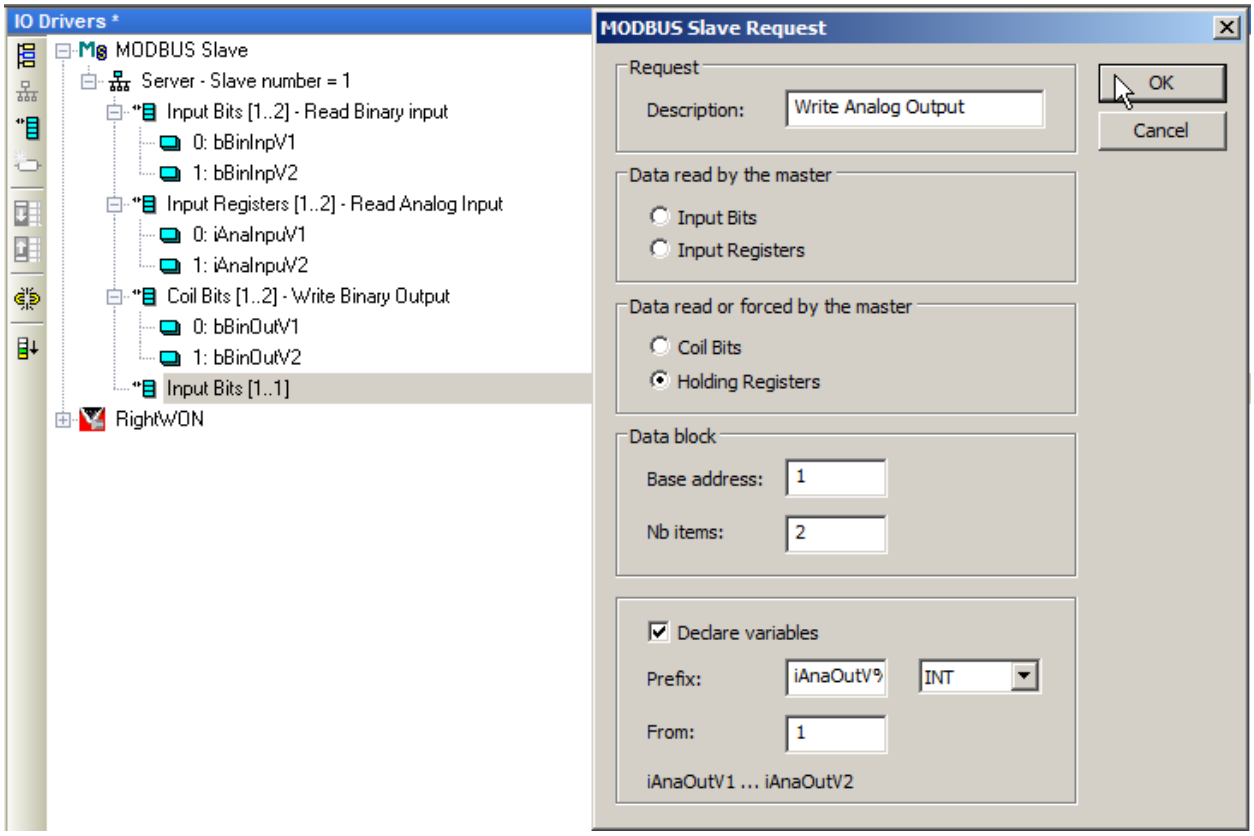
4- Click **OK** to create the data block and the variables.

5- Repeat steps 1 to 3 for each data block to be created. For the purposes of the tutorial, create the data blocks according to the figures below.

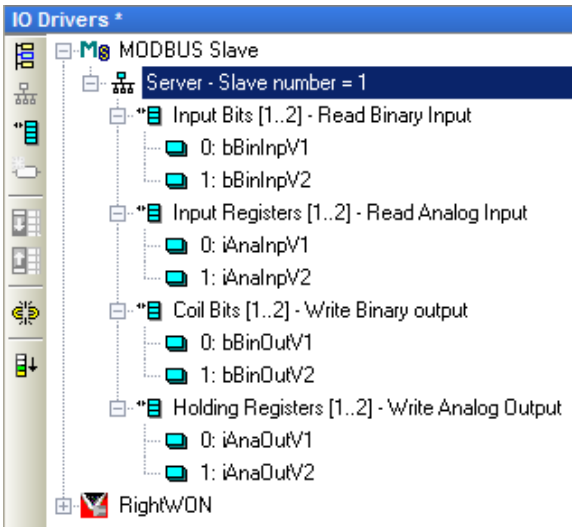








The figure below shows the final result of creating the data blocks and variables.



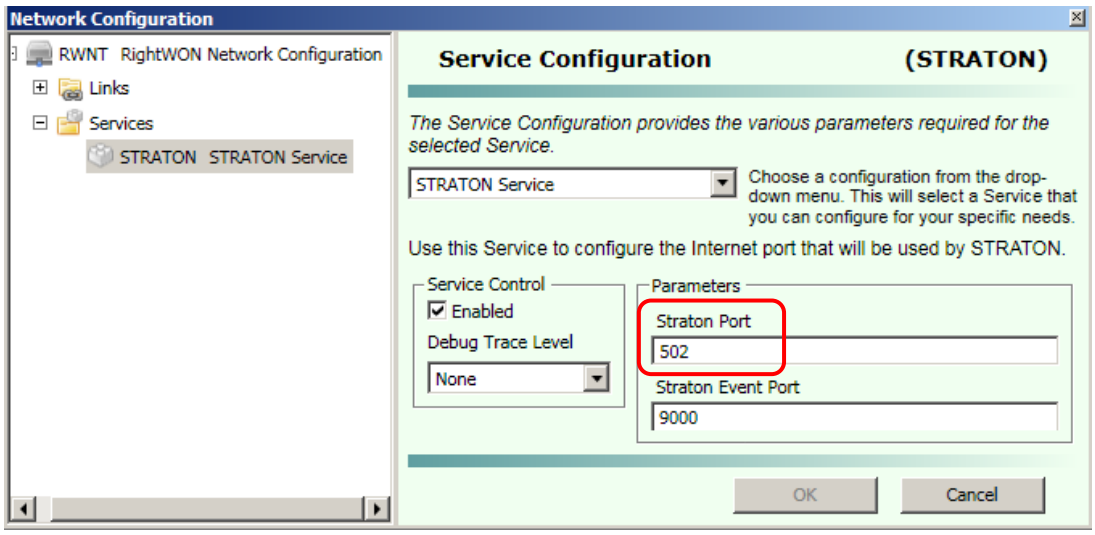
2.4.5. Configuring to open a communication port

To enable the MODBUS Slave to communicate with the MODBUS Master you must configure the application to open an **Ethernet over MODBUS-TCP** or **Serial MODBUS-RTU or Ethernet over MODBUS-UDP** communication port.

2.4.5.1. Configuring to open a **Ethernet over MODBUS** communication port

The default port number for the MODBUS-over-Ethernet port is *502*. This should entered in the **PORT** field.

You must also replace the port number defined in the **Straton Port** field of the STRATON service with 502. To access the STRATON service, double click on **Network** in the **IO Drivers** window, expand the tree view and click on **STRATON Service**.



2.4.5.2. Configuring to open a **Serial MODBUS-RTU** or **Ethernet over MODBUS-UDP** communication port

1- Create a program

2- Configuring the functional block

3- Example of configuring a block and editing a program to open a serial MODBUS-RTU port.

2.4.5.2.1. Creating a program

To create a new program, carry out the following steps:

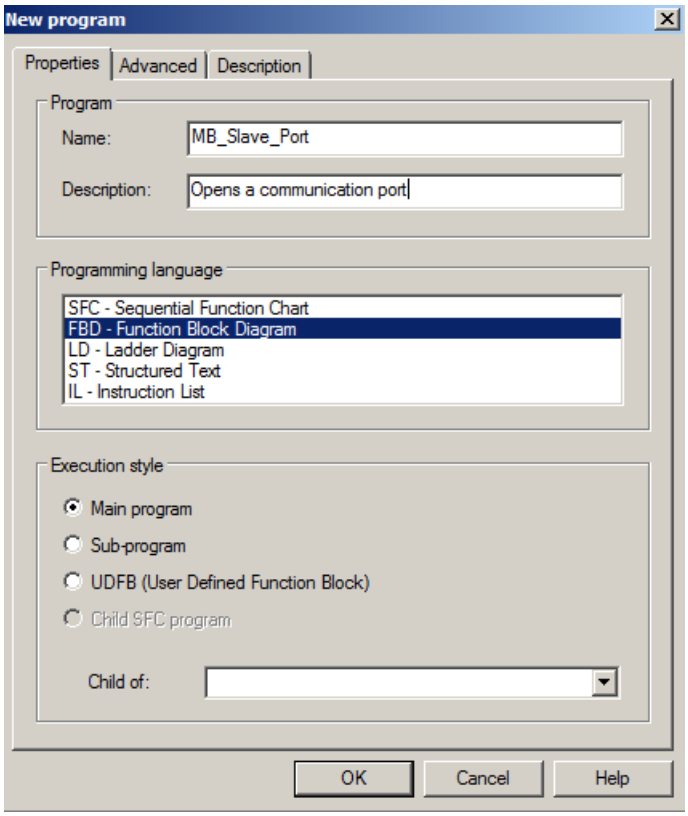
1- In the work area, right click on **Program** and select **New Program...**

2- In the **Name** field, enter the name of the program (e.g. *MS_Slave_Port*).

3- Enter a **Description** of the program (e.g. *Opens a communication port*).

4- Choose **FBD – Function Block Diagram** as the programming language.

5- Click **OK** to insert the program.



2.4.5.2.2. Configuration of functional blocks

To enable MODBUS to communicate with the Master you must configure a program so that the Slave will open a communication port. Configure this port using the function block associated with the protocol variant used:

- **Serial MODBUS-RTU:** Use function blocks **MBSLAVERTU**, **MBSLAVERTUEX**.

  The inputs of the functions are:

  IN: Boolean (BOOL): The port is open when the value is TRUE.

  PORT: Character string (STRING): Enter the logical link associated with the interface.

  SLV: Dual integer (DINT): MODBUS Slave number.

SrvID: Character string (STRING): For MBSLAVERTUEX only, the server ID in the configuration.

The output of the functions is:

Q: Boolean (BOOL): The output is TRUE if the port is successfully opened.

- **MODBUS-UDP over Ethernet:** Use function blocks **MBSLAVEUDP**, **MBSLAVEUDPEX**.

  The inputs of the functions are:

  IN: Boolean (BOOL): The port is open when the value is TRUE.

  PORT: Dual integer (DINT): ETHERNET port number, default: 502 (Note: replace the **STRATON port** number defined in the STRATON Service).

  SLV: Dual integer (DINT): MODBUS Slave number.

  RTU: Boolean (BOOL): Select the UDP protocol: TRUE = MODBUS RTU, and FALSE = Open MODBUS

  SrvID: Character string (STRING): For MBSLAVEUDPEX only, the server ID in the configuration.

  The output of the functions is:

  Q: Boolean (BOOL): The output is TRUE if the port is successfully opened.

2.4.5.2.3. Serial MODBUS port

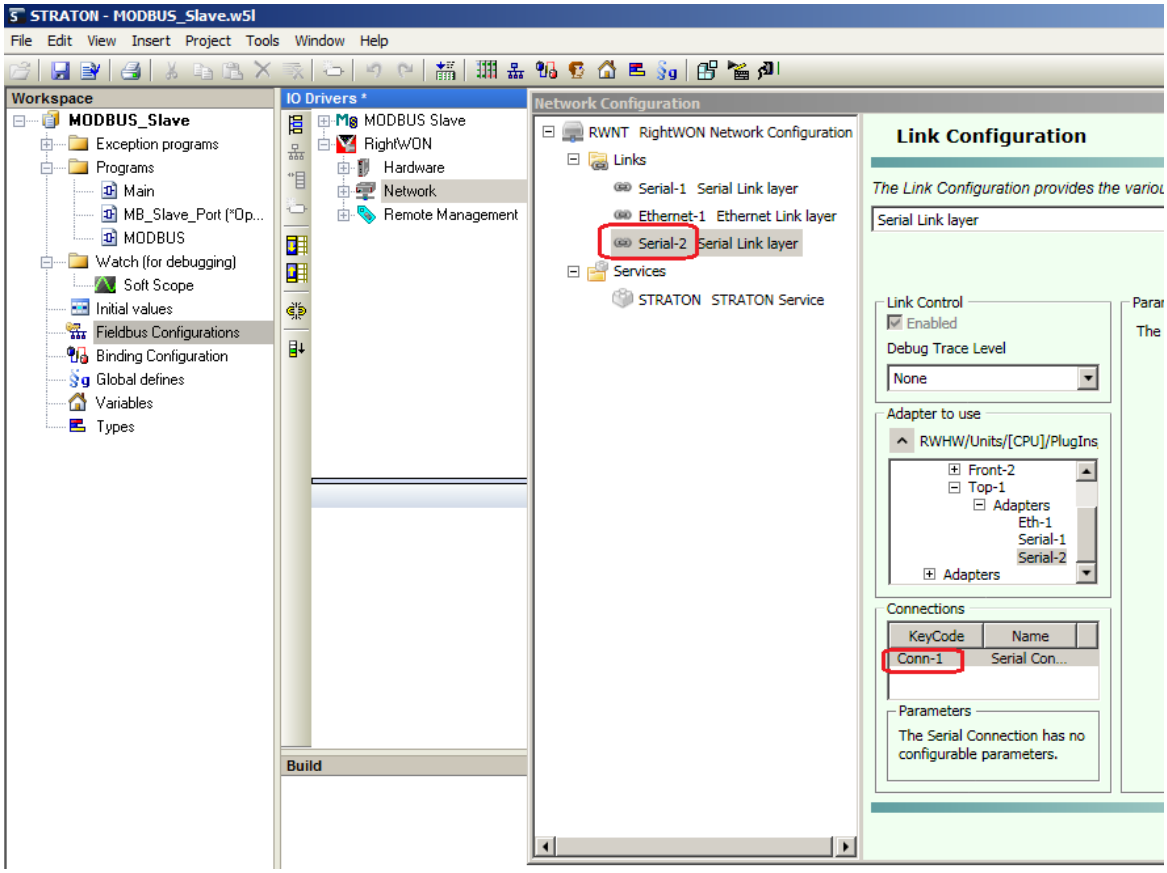
To determine what to enter as the logical link associated with the interface, follow these steps:

1- Double click on **Network Configuration**  , select the **Serial-x** link under **Links.**

2- In the **Adapter to use** section, expand the tree view. Click on **Serial-x** under **Front-1** or **Front-2**, depending on where the serial communication module is installed.

3- Concatenate the **Serial-x** name in the **Links** section with the **Conn-x** name in the **Connections** section.

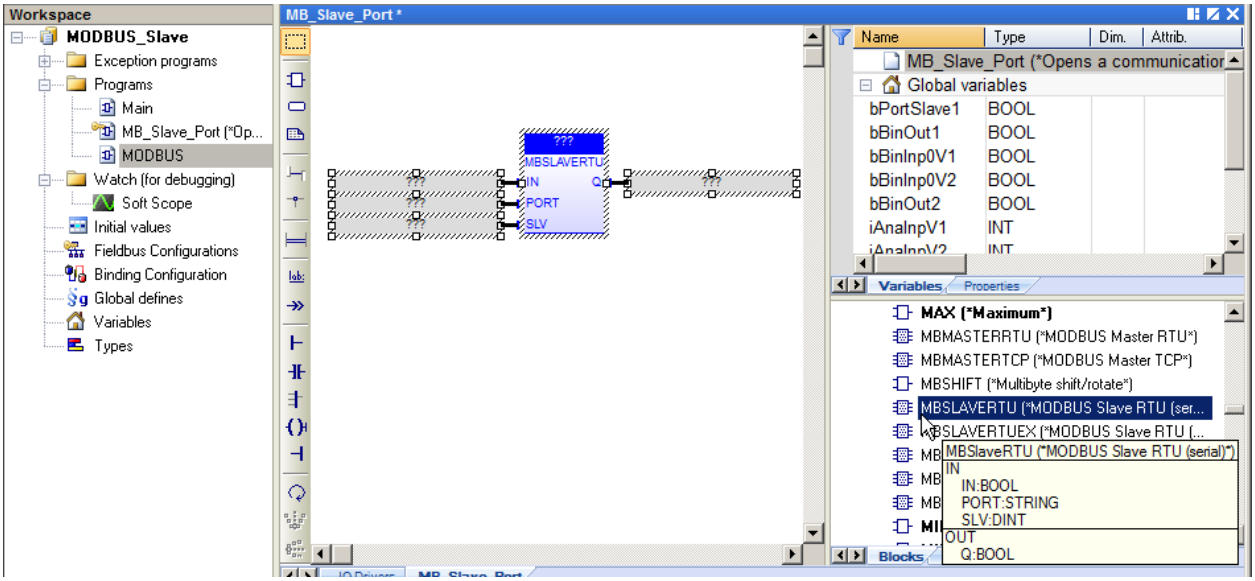
Example of a logical name to enter in the **PORT** field: *'Serial-2.Conn-1'.*



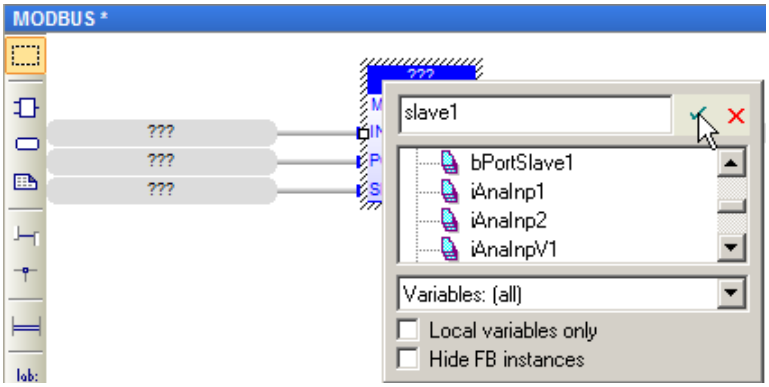
2.4.5.2.4. Configuring and editing a program to open a MODBUS Slave port

In this tutorial the serial MODBUS-RTU protocol is used, so the **MBSLAVERTU** block will be configured to open a communication port with the Master.

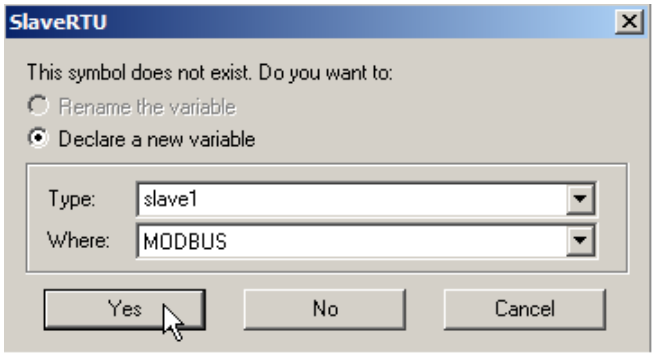
> **Note:** For program editing, refer to the section on editing automation programs in the RightWON in the manual "RWM0000080-MA-en, RightWON Configuration Suite - Application Guide".



1- Click on **???** above the **MBSLAVERTU** block, enter the name of the block and click on √.



2- Click on **Yes** to declare the variable.



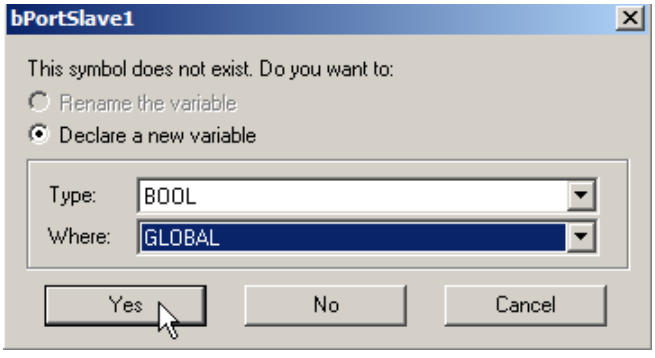
3- Click on **???** to the left of **IN** and enter **true** to activate the block.

4- Click on **???** to the left of **PORT** and enter the logical link associated with the interface. The character string must be within ‘’, e.g. **‘Serial-2.Conn-1’**.

5- Click on **???** to the left of **SLV** and enter the Slave number, e.g. **1**.

6- Click on **???** to the right of **Q** and enter the output name, e.g. **bPortSlave1**.

Choose the **GLOBAL** location and click **Yes** to declare the variable.

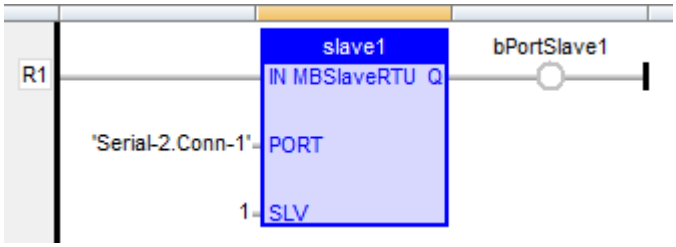


7- Once configured, the MODBUS-RTU block looks like the following figure. The program is now configured to open a communication port with the MODBUS Master.

Example of configuring a functional block in the FBD language

```
slave1 ( TRUE, 'Serial-2.Conn-1', 1 );
bPortSlave1 := slave1.Q;
```

Example of configuring a functional block in the ST language



Example of configuring a functional block in the LD language

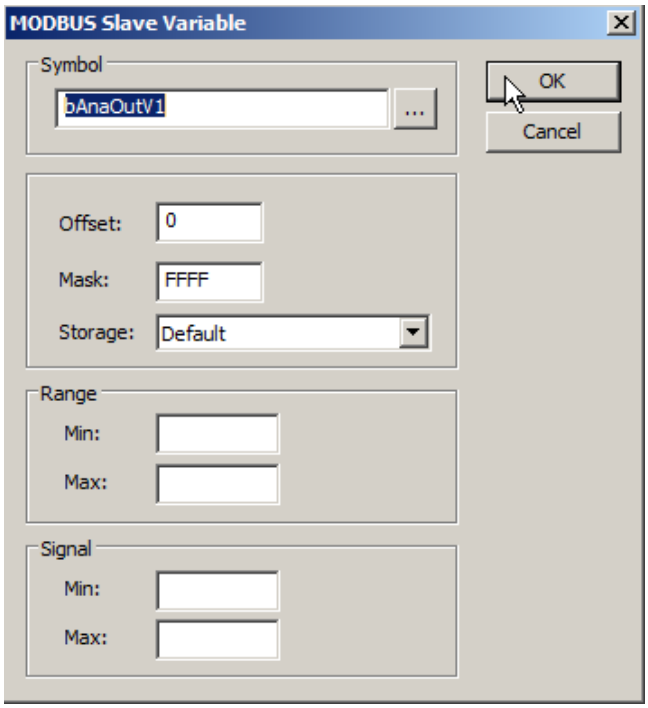
2.4.6. Inserting and configuring a data exchange variable

To transfer variables other than BOOL, INT, UINT, or to specify a mask on a word, carry out the following steps:

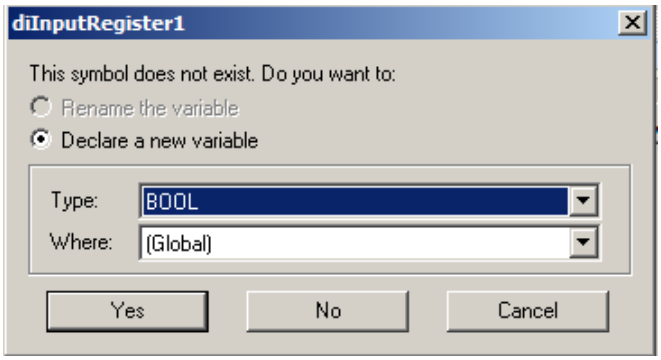
1- When a data block is selected, use the **Edit/New variable** command to link a variable to an item in the block.

2- Each variable is identified by its **Symbol** in the application and an **Offset** from the MODBUS base address of the data block. For further details on masks (**Mask**) refer to the section on how to handle address discontinuities.

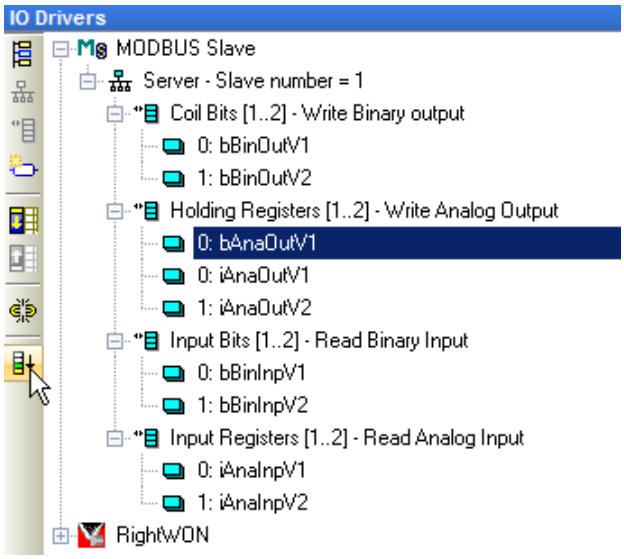
3- Click **OK**.



4- A window opens; choose the **Type** and location (**Where**) of the variable. Verify the data types supported by MODBUS. Click **Yes** to create the variable.



You can sort the variables of each block at any time according to their offset. To do this, select **Server - Slave number** in the **IO Drivers** window, and use the **Edit/Sort symbols** command or click on its icon in the toolbar.



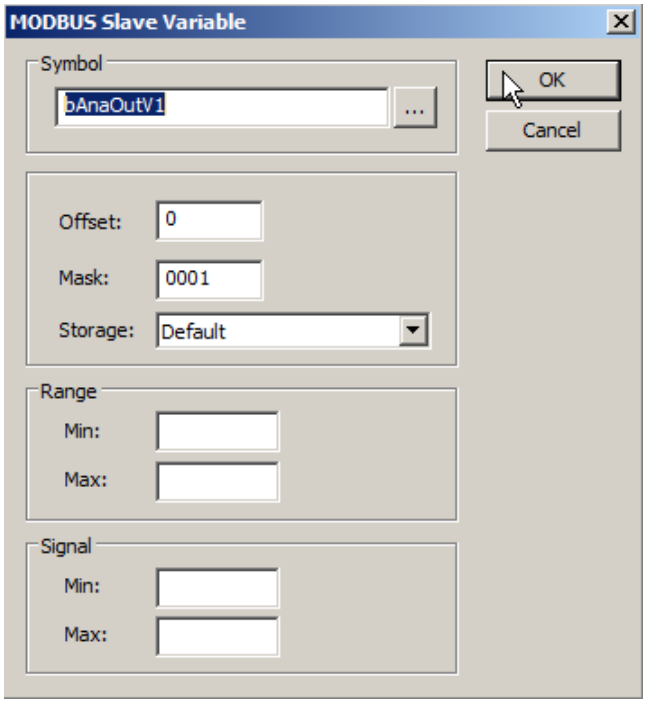
2.4.6.1. How to handle address discontinuities

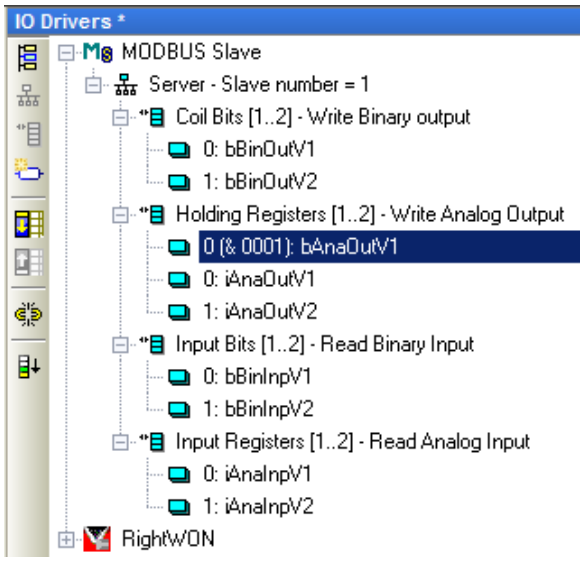
To exchange variables in a MODBUS word, specify a hexadecimal mask that indicates to which bit(s) of the word the variable belongs.

Examples of using three binary masks (Mask) on a word (Word) to allow the exchange of bits in a word:

Word	1	1	1	0	0	0	1	0	1	0	0	0	1	0	1	1
Mask & 0001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bit1																1

When inserting a variable, enter the hexadecimal value of the mask in the **Mask** field.

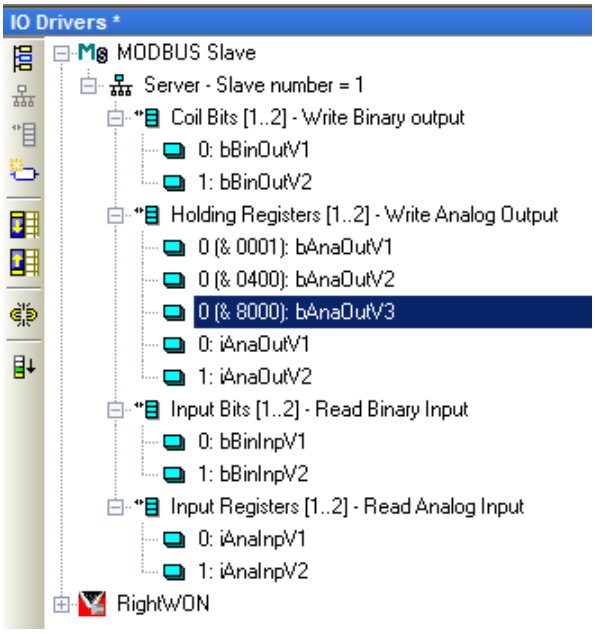




Word	1	1	1	0	0	0	1	0	1	0	0	0	1	0	1	1
Mask & 0400	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Bit2						0										

Word	1	1	1	0	0	0	1	0	1	0	0	0	1	0	1	1
Mask & 8000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit3	1															

After inserting the two other masks, the example should look like the figure below.



2.5. Tutorial on configuring the MODBUS Master

If you have not already done so, you must configure the hardware and the network first. Then follow these steps to configure a MODBUS Master in the RightWON:

1- Manage the status bits associated with the variables.

2- Add the MODBUS Master protocol to the RightWON configuration.

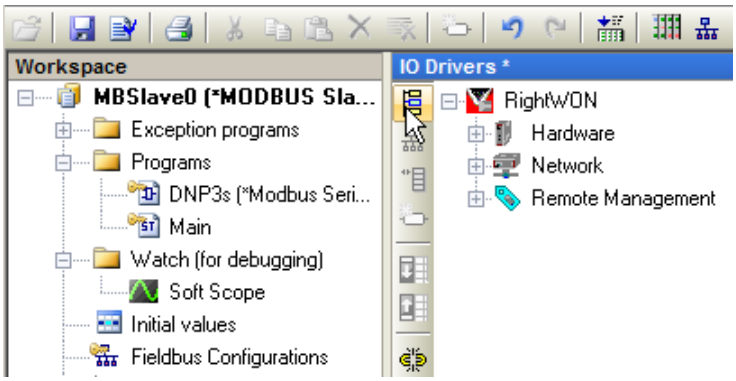
3- Insert and configure a communication port for the MODBUS Master protocol.

4- Add and configure a data block (session), to permit the insertion of variables when configuring the block.

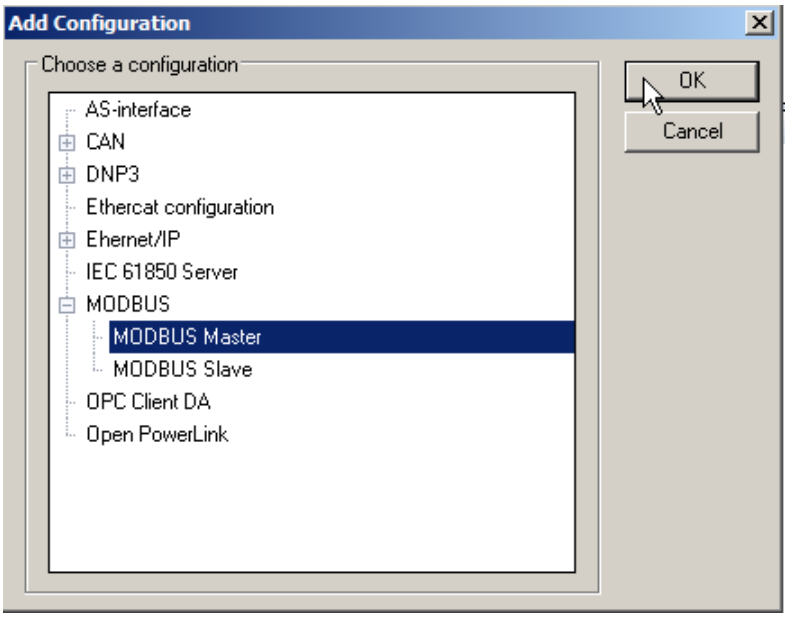
5- Insert and configure a variable in the session for the exchange of data, status or controls.

2.5.1. Adding the MODBUS protocol to the RightWON configuration

1- In the **IO Drivers** window of the **Fieldbus Configurations**, click on the **Insert configuration** icon.



2- Click on the **MODBUS Master** protocol, then click **OK** to insert the protocol.

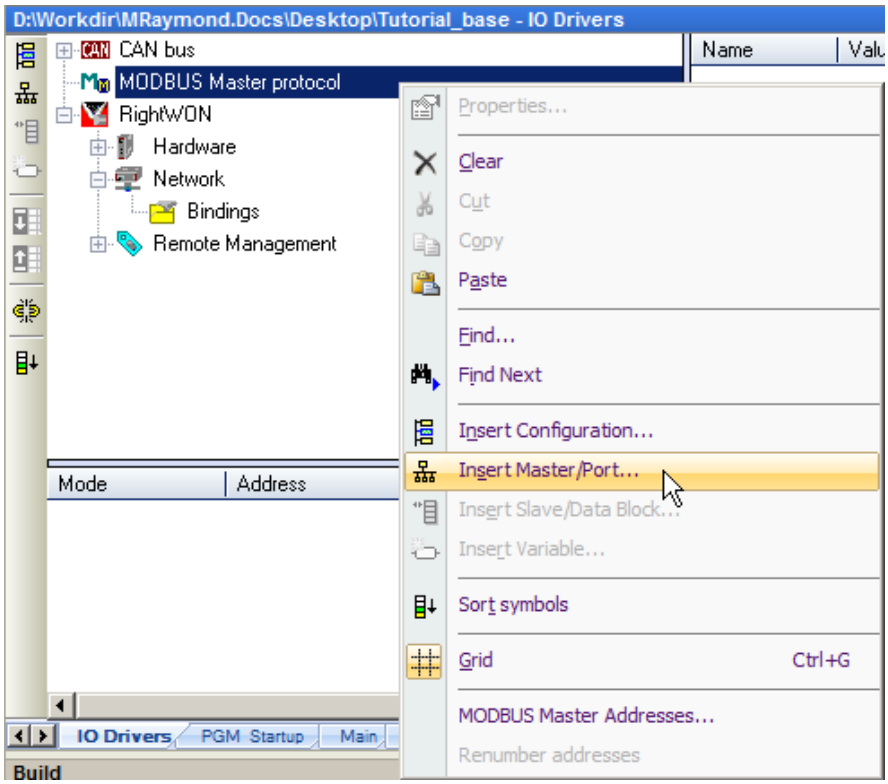


2.5.2. Inserting and configuring a communication port

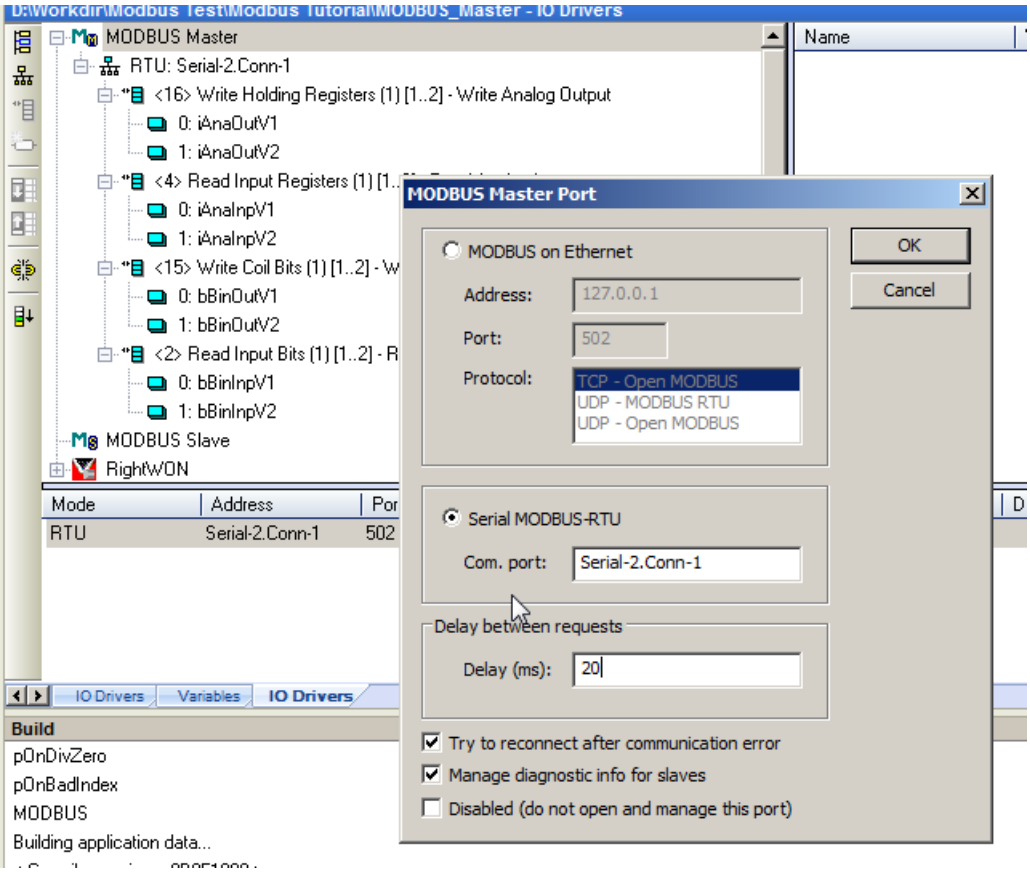
You must configure each MODBUS Master port to access inputs/outputs or a remote system over MODBUS, by executing the following steps:

1- In the **IO Drivers** window, right click on **MODBUS Master**.

2- Click on **Insert Master/Port...** 🖧.

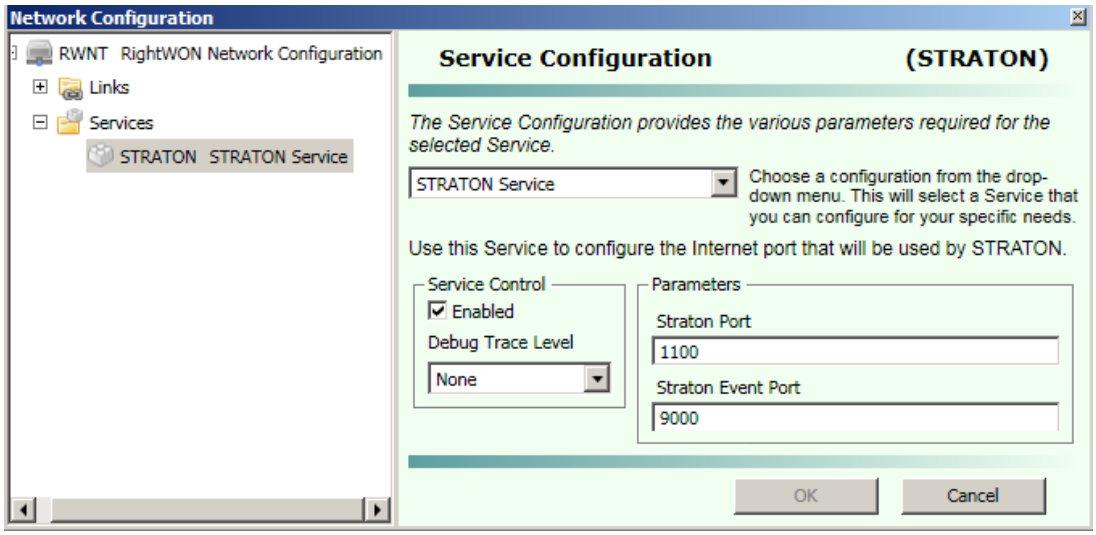


3- Each port identifies a physical and logical MODBUS port. A port is based on a serial or Ethernet link defined in the network configurator. Select one of the MODBUS protocol variants for communication with the Slave, and ensure that the Master and Slave are configured with the same variant.



a. For the **MODBUS over Ethernet** protocol:

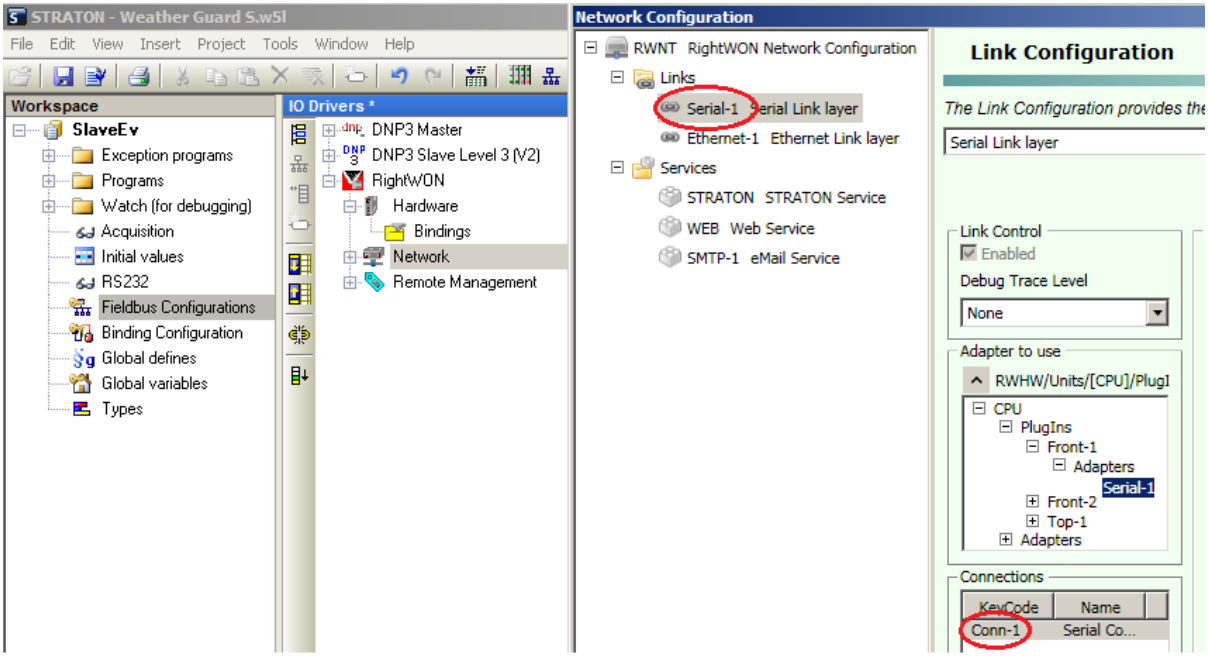
- In the **Address** field, enter the IP address of the Slave, which is the **IP Local Address** of the Ethernet link for the MODBUS Slave.
- In the **Port** field, enter the number of the Ethernet port used by the Slave. The default port is 502. If the protocol variant is **TCP-Open MODBUS**, the port is the one defined in the **Straton Port** field of the STRATON service (default port: 1100). To access the STRATON service, double click on **Network** in the **IO Drivers** window, expand the tree view and click on **STRATON Service**.



- Select the protocol variant to be used, which can be **TCP- open MODBUS**, **UDP – MODBUS RTU** or **UDP – open MODBUS**.

b. For the **serial MODBUS RTU** protocol, enter the logical link associated with the interface; see the note below.

Note: Open the **Network Configuration** , select the **Serial-x** link under **Links**. In the **Adapter to use** section, expand the tree view. Click on **Serial-x** for **Front-1** or **Front-2**, depending on where the serial communication module is installed. Concatenate the **Serial-x** name in the **Links** section with the **Conn-x** name in the **Connections** section (e.g. *Serial-1.Conn-1* in the figure below).



For example, check **Serial MODBUS-RTU** and enter *Serial-1.Conn-1* in the **Com. port** field.

c. **Delay between requests (ms):** Enter the value of the delay.

d. **Try to reconnect after communication error:** Check this field to specify that the Master should try to reconnect after losing communication or on communication error.

e. **Manage diagnostic info for slaves:** Check this field to use diagnostic information from the Slaves (status variables).

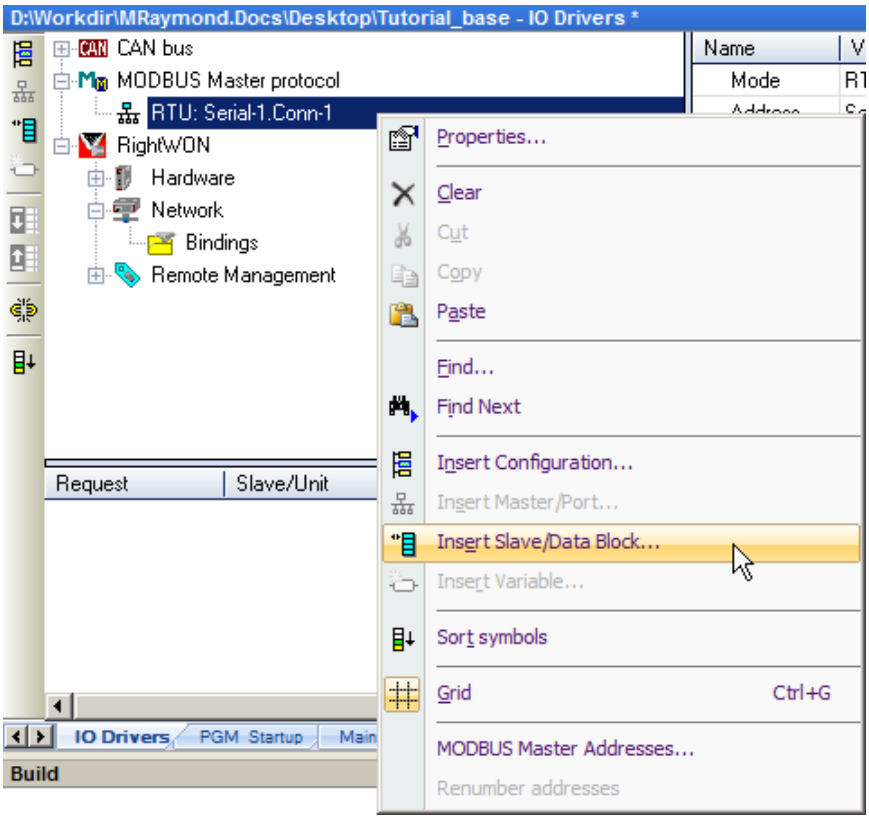
f. **Disabled:** Check this field if this port is not to be used.

4- To insert the port, click **OK**.

2.5.3. Inserting and configuring a data block (request)

When a port is selected, define the MODBUS data blocks by carrying out the following steps:

1- Under **Fieldbus Configurations**, right click on the communication port (e.g. **RTU: Serial-1.Conn-1**). Then click on **Insert Slave/Data Block...**.



2- The **MODBUS Master Request** window opens for configuration of the data block:

a. In the **Request** section:

- **Description:** Enter a description of the request (e.g. *Write Analog Output S1*).
- **Slave/Unit:** Enter the Slave number (e.g. *1*).

b. In the **MODBUS Request** section, choose one of the MODBUS functions from the dropdown menu (e.g. *<16> Write Holding Registers*).

c. In the **Data block** section, each block is identified by a **Base address** and number of items, **Nb items** (bits or words). The number of items is limited by MODBUS (2000 bits read, 1968 bits forced, 125 words read or 120 words forced). (E.g. Base Address: 1, Nb items: 2).

⚠ **Caution!** Read/write requests sent from MODBUS Masters will be refused by the MODBUS Slave if the address range specified in the request is not included in any of the blocks configured on the MODBUS Slave. Requests that intersect several blocks configured on the MODBUS Slave will also be rejected. For example, if you configure a 16-word block at address 1 and another 16-word block of at address 17, a 32-word read or write request starting from address 1 will be refused and an "invalid address" report will be returned.

d. In the **Activation** section, click on one of the possible activation modes:

- **Periodic:** The request (data block) is sent to the Slave according to the specified time period (ms). You must also specify the time period (**on error**) to be used when the request is in error. This is typically used to slow down exchanges with the Slaves when a communication error occurs.
- **On call:** The request is sent to the Slave when the control variable changes to TRUE in the program. See the configuration example for sending a request on call.
- **On change:** Used for a write request only, indicates that the request is sent only when at least one of the variables in the request changes state.

(E.g. check **Periodic**, enter **0** ms and **0** (on error).)

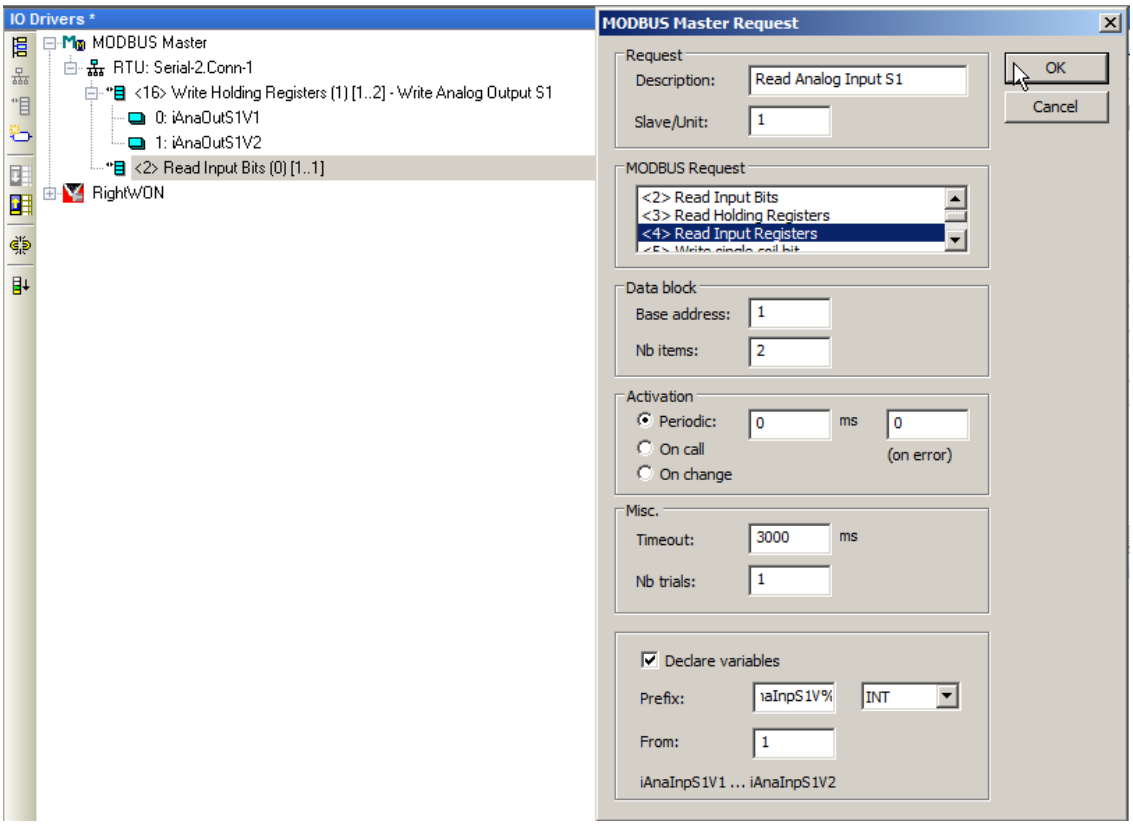
e. In the **Misc.** section, enter the wait time before a communication failure, **Timeout (ms)**. Enter the number of reconnect attempts (**Nb trials**) after a communication error with the Slave. (E.g. Timeout of 3000 ms and 1 attempt.)

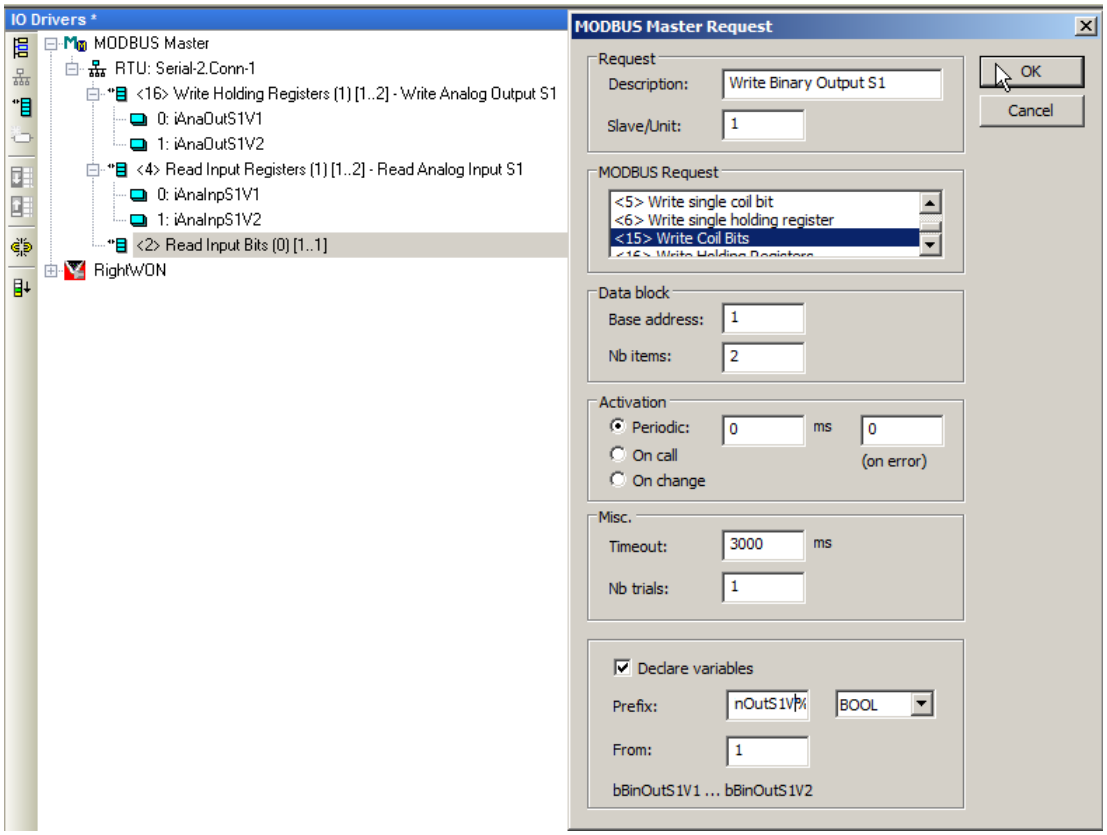
f. To insert the variables in the data block, check the **Declare variables** box. Enter the prefix of the variables in the **Prefix** field followed by **%** (e.g. **V%**). For a word-based request, choose the variable type (e.g. **INT** or **UINT**) from the dropdown menu. This will automatically create the variables in the data block and in the global variables according to the **Base address** and number of items (**Nb items**).

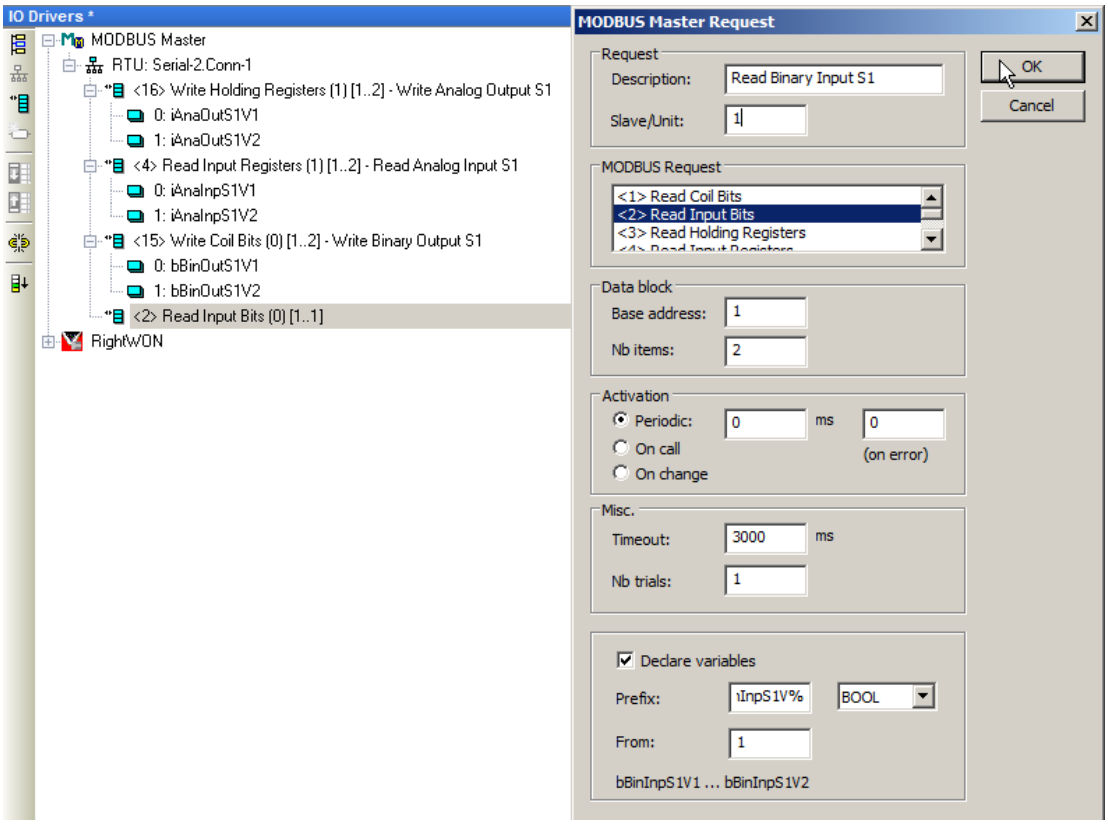
Note: To transfer variables other than BOOL, INT and UINT, or to specify a mask on a word, uncheck **Declare variables** and manually add the variables to the data block.

3- Click **OK** to create the data block and the variables.

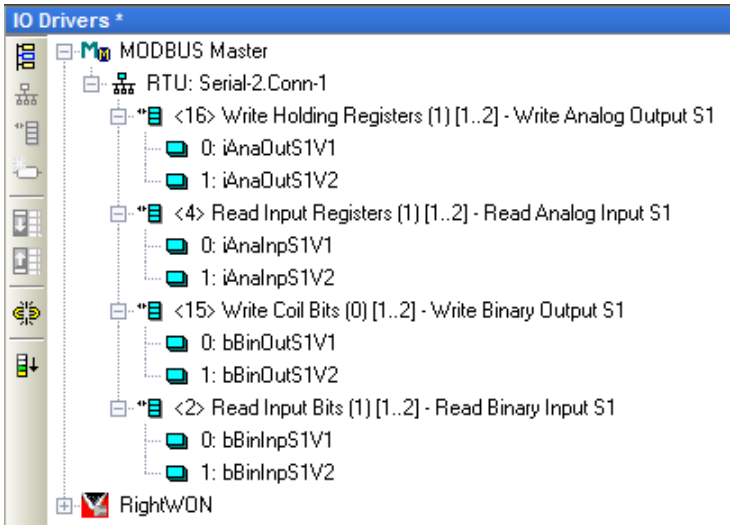
4- Repeat steps 1 to 3 for each data block to be created. For the purposes of the tutorial, create the data blocks according to the figures below.







The following figure shows the final result of creating the data blocks and variables.



2.5.4. Inserting and configuring a variable for data, status or control exchange

When a data block is selected, use the **Edit/New variable** command to link a variable to a data block. You can choose from:

- A data exchange (between the MODBUS stack and an application variable), to transfer variables other than BOOL, INT and UINT, or to specify a mask on a word.
- A status (to provide a diagnostic to the application) or control (to send a request on call or manage the status counters).

To configure a status or control variable, carry out the following steps:

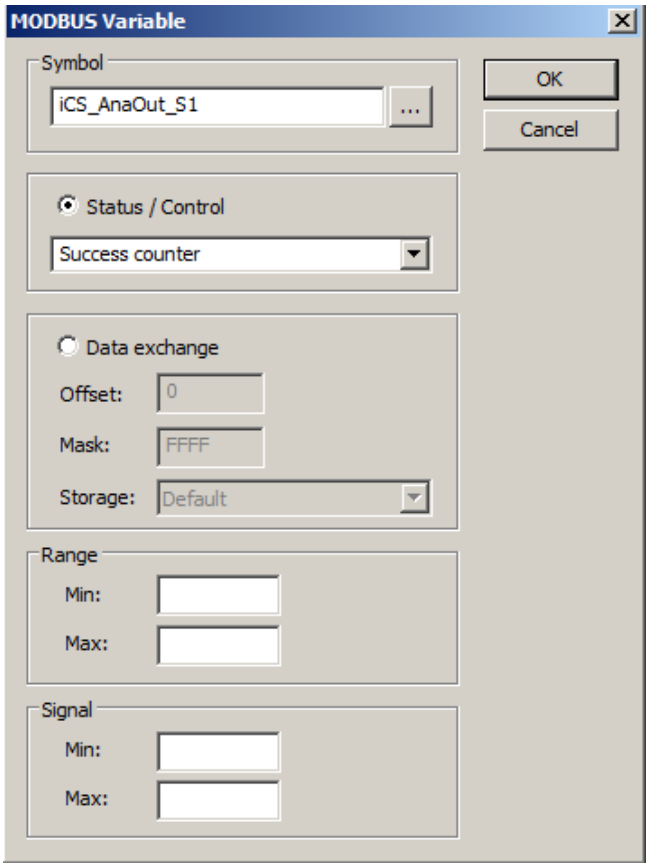
Note: To configure a *Command (one shot)* or *Command (validation)* variable, refer to the section on how to send a request on call.

1- As an example, right click on the **<16> Write Analog Output S1** request, then click on **Insert variable...**

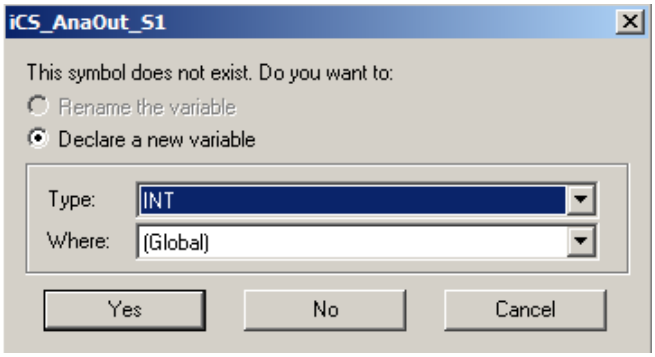
2- In the **Symbol** field, identify the variable (e.g. *iCS_AnaOut_S1*).

3- From the **Status/Control** dropdown menu, select the type of status or control variable, (e.g. **Success counter**).

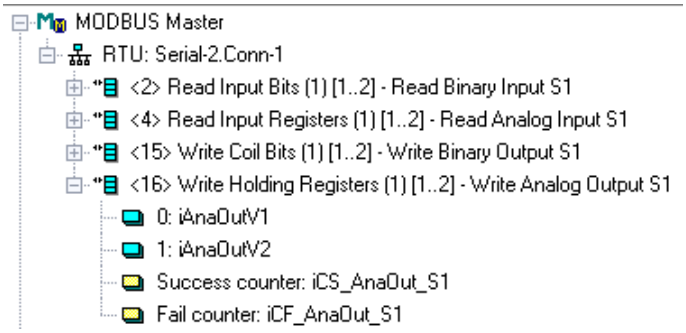
4- Click **OK**.



5- Choose the **Type** of variable (e.g. **BOOL**, **INT**, **UINT**) and click **Yes** to declare the variable (e.g. choose the **INT** type).



6- The figure below shows the result of creating the status variables.



2.5.4.1. MODBUS Master status and control variables and error codes

A status is a diagnostic that is provided to the application. Status variables can return error codes. A control permits using a variable to drive the stack.

The available types of diagnostic and status information are:

Information	**Description**
Error report	The variable is forced to an error code after the last unsuccessful attempt, and is reset after the first successful retry.
Error report (Set only)	Identical to the above, except that the variable is not reset by the MODBUS stack. The application handles the reset of the variable.
On-going request	The variable is set to 1 when the request is issued, and is reset after the Slave responds or a timeout expires.
Success counter	The variable is incremented after each successful exchange.
Fail counter	The variable is incremented after each failed exchange.
Retry counter	The variable is set to 0 on the first attempt, and is incremented for each retry after a failure.
Slave: last error	The variable contains the code for the last error that occurred for this Slave.
Slave: last error date stamp	The variable contains the date of the last error that occurred for this Slave.
Slave: last error time stamp	The variable contains the time of the last error that occurred for this Slave.
Slave: last reset date stamp	The variable contains the date of the last reset of the diagnostic counters for this Slave.
Slave: last reset time stamp	The variable contains the time of the last reset of the diagnostic counters for this Slave.
Slave: transaction counter	The variable is incremented for each new exchange with this Slave, whether or not the exchange is successful.
Slave: failed transaction counter	The variable is incremented for each new error during an exchange with this Slave.

These status variables can return error codes.

The following controls can be configured:

Control	Description
Command (enable)*	The request is sent continuously while the variable is TRUE.
Command (one shot)*	The request is sent only once when the variable changes to TRUE. The variable is reset at the end of the exchange.
Reset counters	Diagnostic counters for the request are reset when the variable changes to TRUE, and then the variable is reset.
Slave: reset counters	Diagnostic counters for the Slave are reset when the variable changes to TRUE, and then the variable is reset.

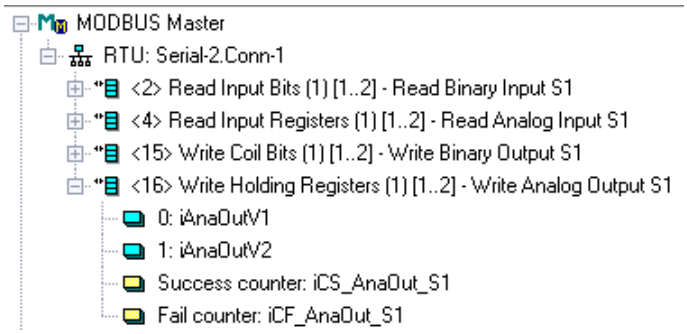
* In specifying this control option, the data block will be exchanged only when the variable is in the TRUE state (or non-zero for an integer).

The possible error codes for the status variables are:

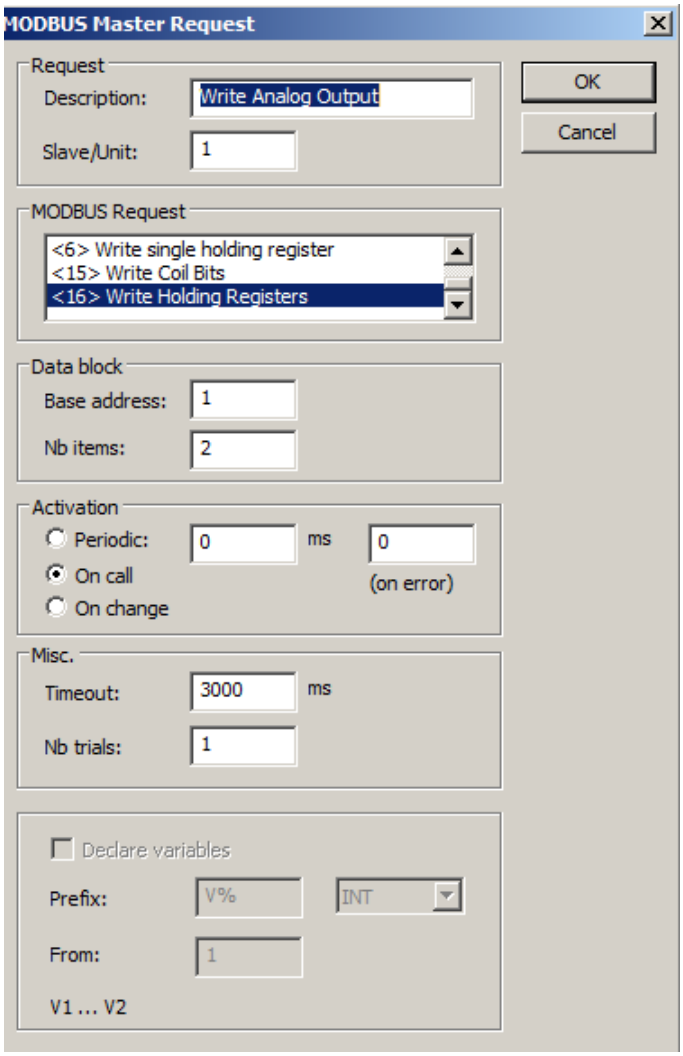
Error code	Description
0	No error
1	MODBUS function not supported
2	Invalid address
3	Invalid MODBUS value
4	MODBUS server error
5	Server acknowledge
6	Server busy
8	Parity error
10	Invalid gateway routing
11	Wrong gateway target
128	Communication timeout
129	Incorrect CRC16
130	Ethernet/serial communication error
200	Invalid Slave number
201	Invalid MODBUS address
202	Invalid item number or data table too small
203	Invalid type for data table
204	Invalid number of items

2.5.5. Example of sending a request on call

In this example, the **<16> Write Analog Output S1** request will be modified to be sent on call:



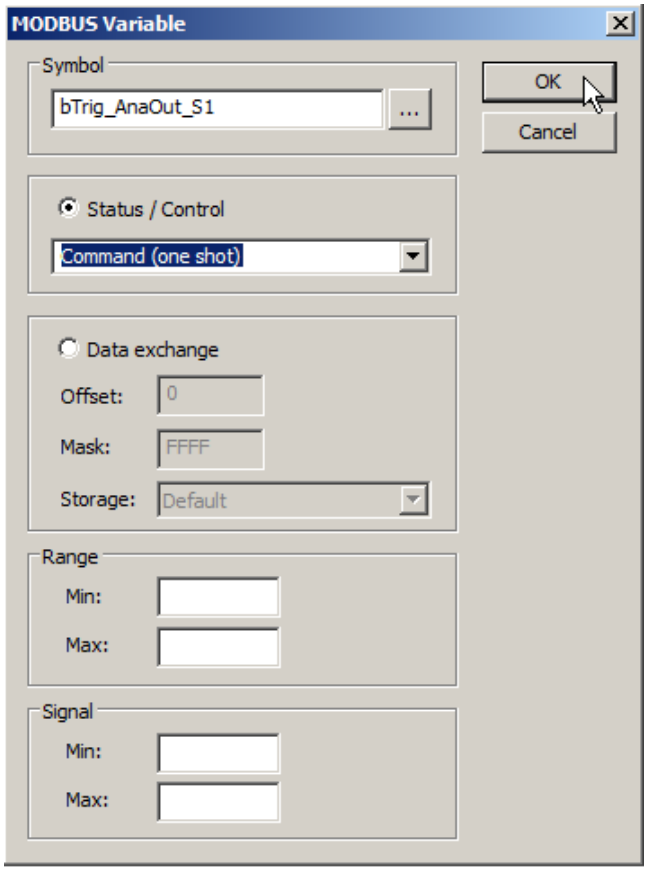
1- Double click on the **<16> Write Analog Output S1** request. The data block must be configured in **Activation - On call** mode.



2- Right click on the **<16> Write Analog Output S1** request, then click on **Insert variable...**

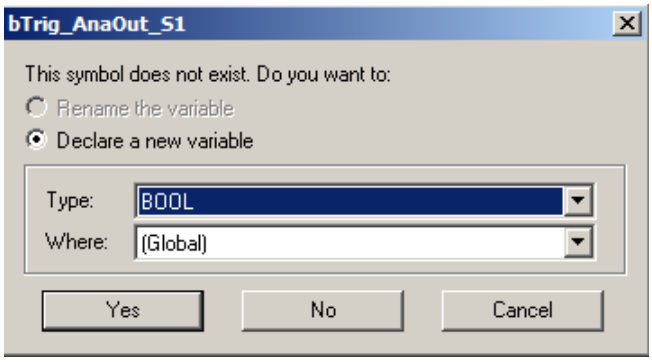
3- Enter a name for the variable in the **Symbol** field (e.g. *bTrigWAnaOut)*.

4- From the **Status/Control** dropdown menu, choose for example *Command (one shot)* (or *validation*).

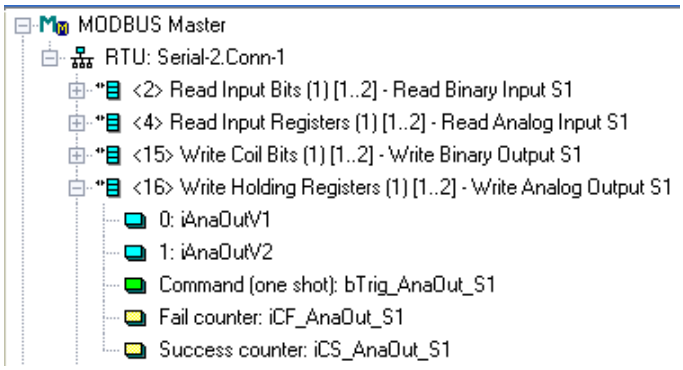


5- Select *BOOL* as the variable **Type**, then click **Yes** to create the variable.

Note: The data block is changed only when the variable is in the TRUE state (or non-zero for an integer).



6- In this example, the control variable is added to the data block. Make sure that the data block is configured in **On call** mode.

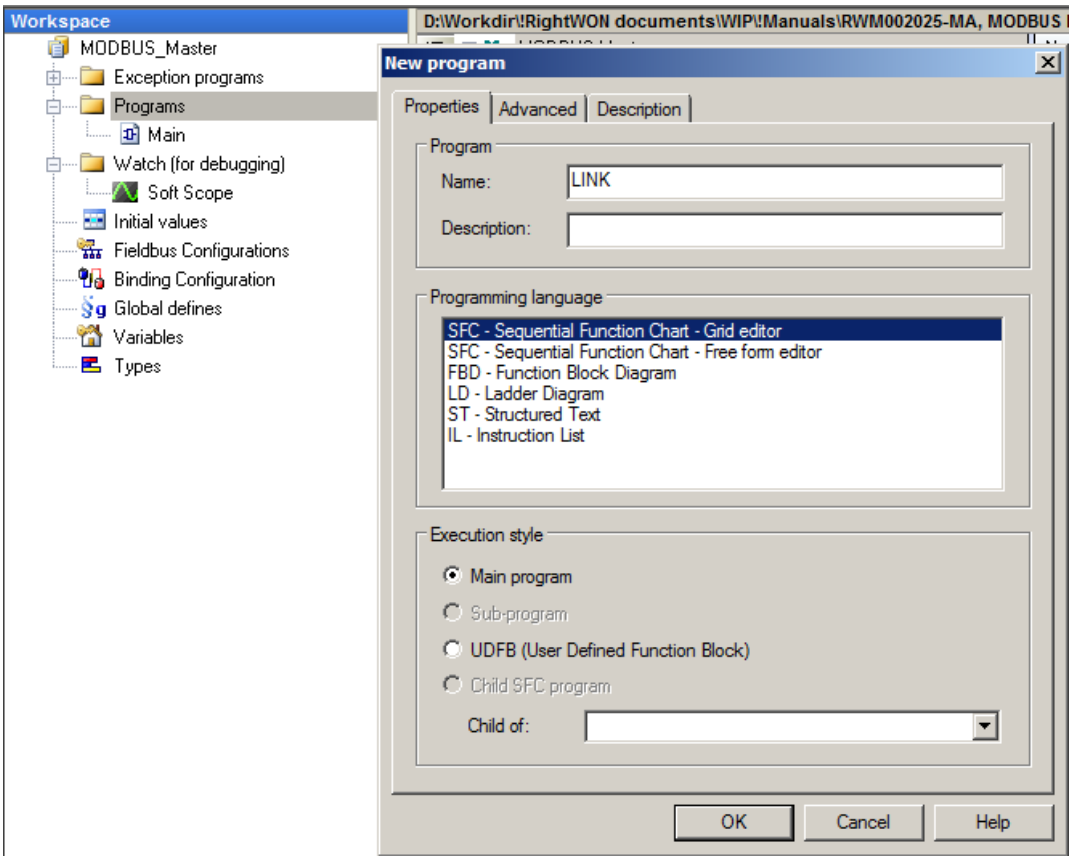


7- Create a program that permits setting the *bTrig_AnaOut_S1* variable to **TRUE** for the Command (one shot).

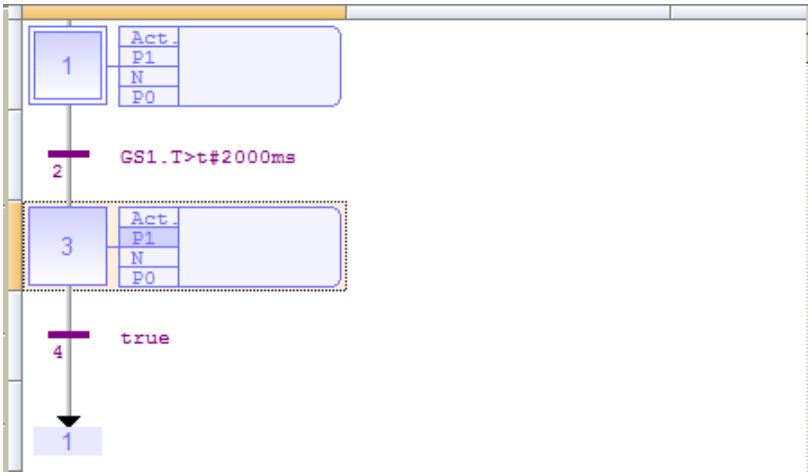
Note: When the value of *bTrig_AnaOut_S1* changes to **TRUE** (BOOL) or **non-zero** for numeric types (INT, UNIT, etc.), the *Write Analog Output* request is sent to the Slave. After the exchange of data, *bTrig_AnaOut_S1* automatically returns to **FALSE** or **0**.

For example, create an SFC (Sequential Function Chart) program that sets the *bTrig_AnaOut_S1* variable to **TRUE** every 2 seconds:

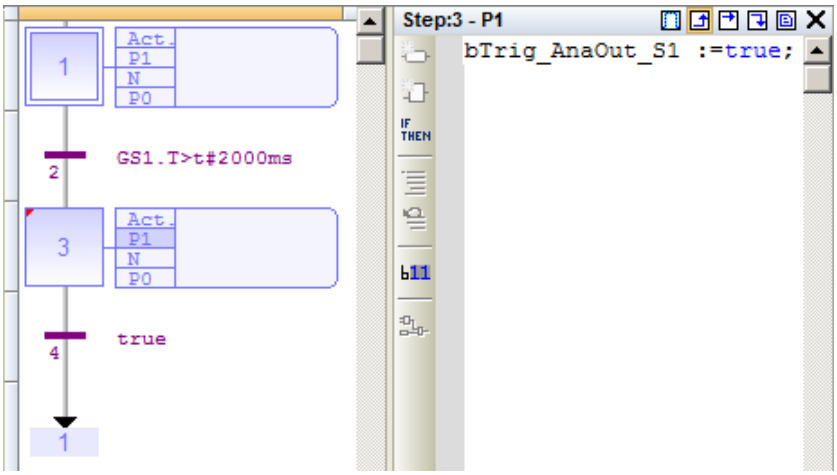
1- Right click on **Programs** in the work area. Enter a **Name** for the program. For the **Programming language** field, choose "SFC – Sequential Function Chart – Grid editor". Click on **OK**.



2- Double click on the LINK program and create the following SFC:



3- Double click on **P1** at step **3**, and enter the following line:



Vizimax Inc.
2284 de la Province
Longueuil, Québec
Canada J4G 1G1

Tel. (450) 679-0003
Fax: (450) 679-9051
Sales@vizimax.com
www.vizimax.com