

## Smart Application tutorial

### **Introduction:**

Smart Application is simple supervision software designed for smart relays controllers. It have been tested with millennium 3 (Crouzet) controllers using standard serial, Bluetooth, and Ethernet link (with XN05 extension module), and with Advantys OTB distributed I/O using Ethernet link.

### **IMPORTANT INFORMATIONS:**

This development is realized totally outside of any professional activities and with no support of Crouzet ©. Crouzet© or developer can't be responsible for any damages or miss-functioning that occur through the use of this free software witch is developed only under privates activities.

This document will teach you how to use this software and supervise M3 controllers through serial (or Bluetooth) and Ethernet link

Equipments required:

- 1 Millenium 3 controller (it must be extensible like XD10(S) 24 VDC or XD26(S) 24VDC for Ethernet use only)
- 1 Ethernet extension XN05 for Ethernet use only.
- 1 computer with Microsoft .Net framework©.

NOTE: .Net framework is downloadable [here](#).

- 1 Millenium 3 Bluetooth key for Millenium 3 controllers (part number 88 970 104 0625) for Bluetooth use only.
- 1 serial link cable for millennium 3 controller for serial link use only.
- CLS M3 2.1.0 software or further versions.

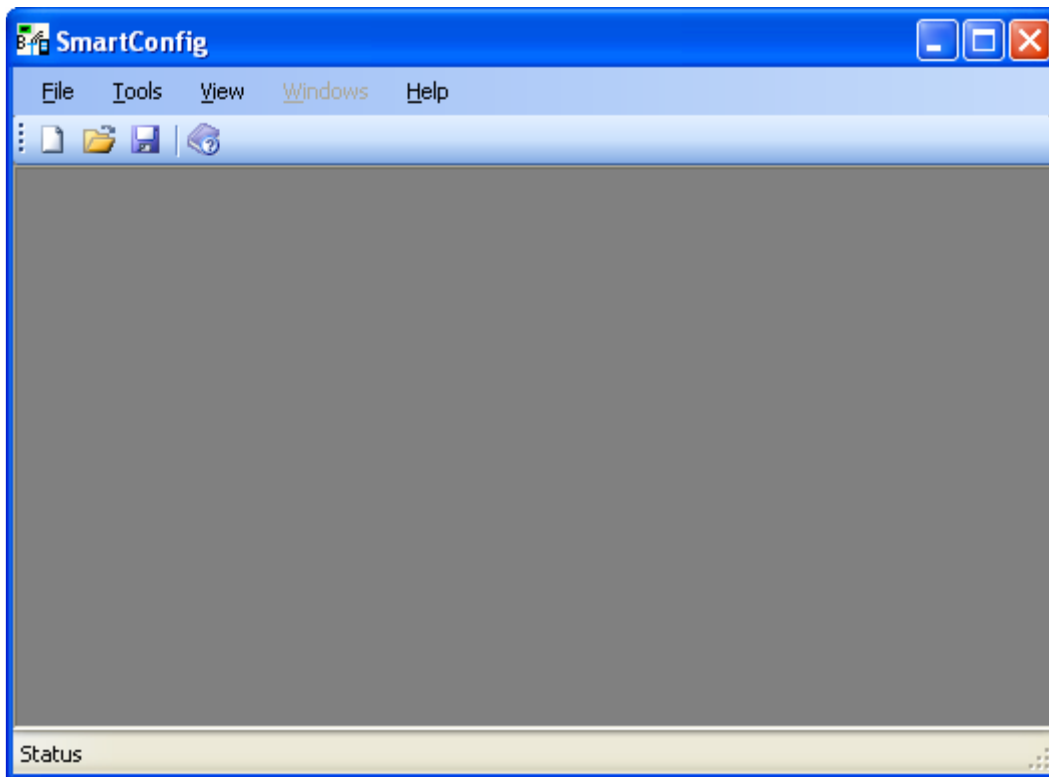
### **Setup:**


Launch Setup.exe and follow instructions. Once setup is complete, two shortcuts should have been created on your desktop, one for SmartConfig and one for SmartCommand

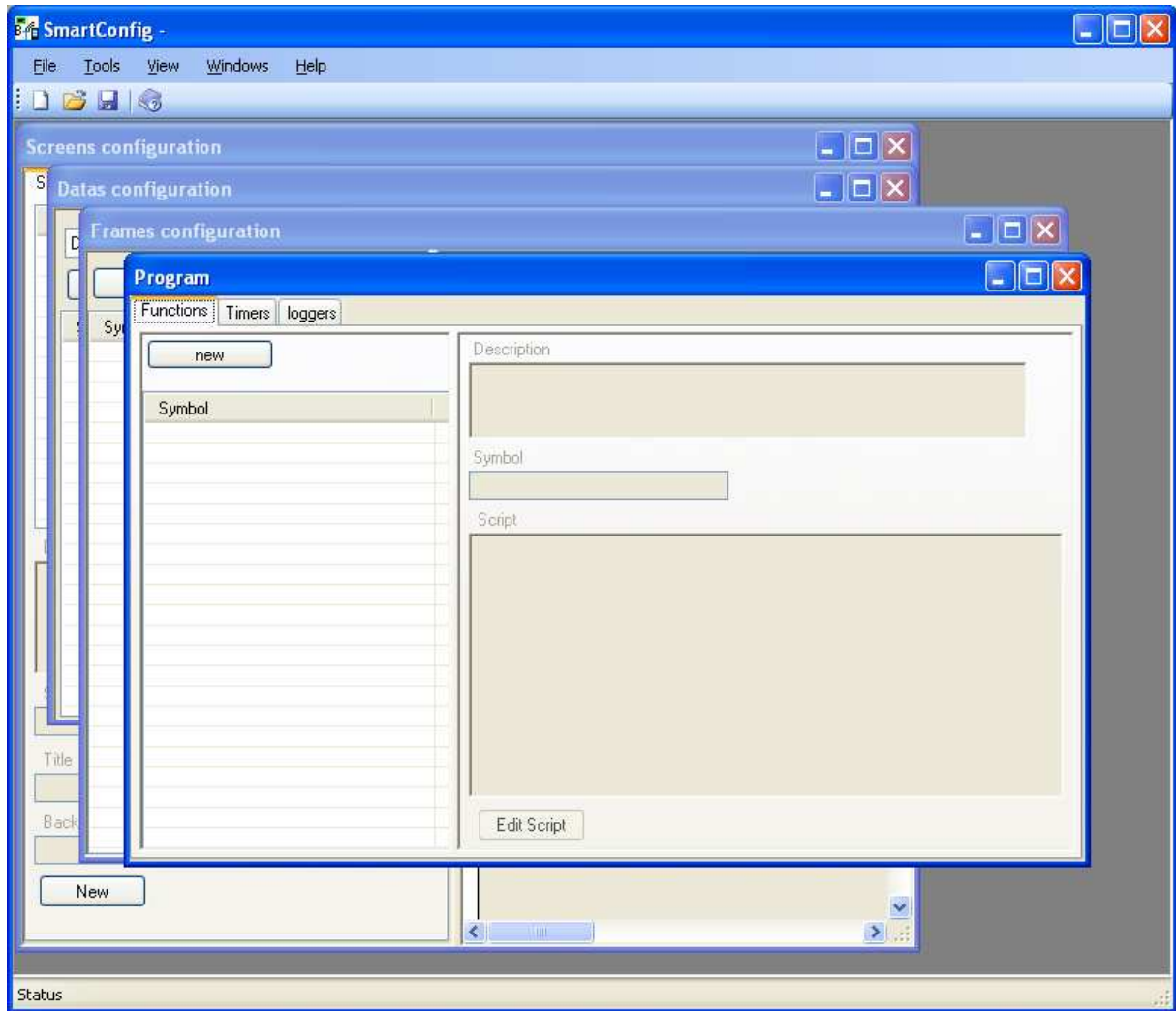
1. Creating a supervision configuration file .....	3
2. Frames Wizards .....	4
2.1 M3 SL Bloc Wizard. ....	5
2.2 TCP Modbus Wizard.....	9
3. Frame Window .....	12
3.1 Frame properties.....	13
3.2 Frame datas .....	15
4. Data Window.....	16
5. Screen Window .....	19
5.1 Screen Page .....	20
5.2 Designer .....	21
5.3 Control Options Page .....	23
Associating datas and controls .....	25
5.4 Item Script Page .....	28
6. Script .....	29
7. SmartCommand.....	32
7.1 Connection configuration.....	33
7.2 Runnig program.....	33
8. Finalize M3 Controller Supervision .....	34
9. Program Window .....	35
9.1 Timers Page.....	36
9.2 Loggers Page .....	36
9.3 Functions Page .....	36

# 1. Creating a supervision configuration file

Start the Smart Configuration (SmartConfig shortcut) software, the following screen appears.



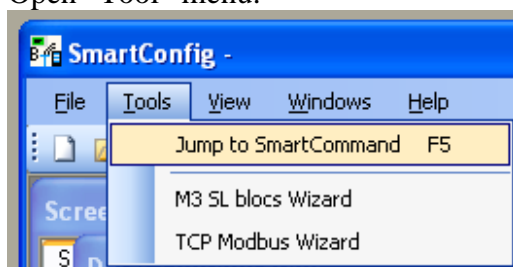
Once the software is started, press the “New” button  or “New” item in “File” menu. 4 new windows appear in configuration software.



Before describing the purpose of each window, we will see how to use frame generator wizard.

## 2. Frames Wizards

Open "Tool" menu.



In this menu, you can find 3 commands:

- Jump to SmartCommand: it allows you to jump directly to SmartComand Application and test currently edited configuration file.
- M3 SL bloc Wizard: this wizard will help you to create exchanged frames for M3 SL BLOC supervision using serial link (or Bluetooth).
- TCP Modbus Wizard: this wizard will help you to create TCP/Modbus frame for M3 XN05 Ethernet supervision.

## 2.1 M3 SL Bloc Wizard.

If you want to use this software through Ethernet using XN05 extension module, jump to [TCP Modbus Wizard](#)

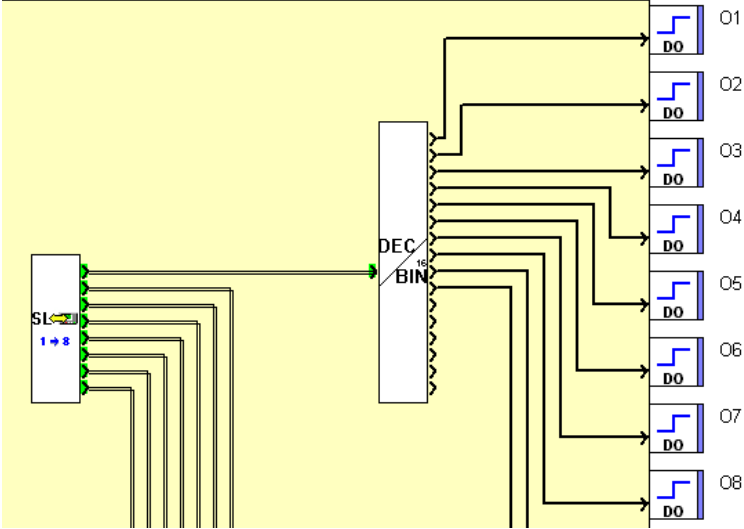
Open M3 SL Bloc Wizard. The following window appears.

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_01	16	1
M3_SLIN1_02	16	2
M3_SLIN1_03	16	3
M3_SLIN1_04	16	4
M3_SLIN1_05	16	5
M3_SLIN1_06	16	6
M3_SLIN1_07	16	7
M3_SLIN1_08	16	8

The list allows you to split each data to smaller size and at least create one data for each bit, quartet of byte for each SL Bloc I/O.

But see how it can be used concretely with **Serial Example (SL Bloc).pm3** file.

Here is a screen shot of M3 Input (address 1-8) SL bloc wiring in sample file.



The first bloc output is connected to a DEC/BIN bloc that allows us to use each bit to order controller outputs. So we want to use each bit of first SL Bloc output as independent data.

Click the first data named M3\_SLIN1\_O1 and then click “Split selected data”.

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_O1	16	1
M3_SLIN1_O2	16	2
M3_SLIN1_O3	16	3
M3_SLIN1_O4	16	4
M3_SLIN1_O5	16	5
M3_SLIN1_O6	16	6
M3_SLIN1_O7	16	7
M3_SLIN1_O8	16	8

Split selected data      Join selected datas

The data is splitted into two 8 bits datas witch are defined on the same output.

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_01_MSB	8	1
M3_SLIN1_01_LSB	8	1
M3_SLIN1_02	16	2
M3_SLIN1_03	16	3
M3_SLIN1_04	16	4
M3_SLIN1_05	16	5
M3_SLIN1_06	16	6
M3_SLIN1_07	16	7
M3_SLIN1_08	16	8

Each created data take a specific name according to its size and IO of SL Bloc.

Here, the two datas are named M3\_SLIN1\_01\_MSB (MSB for Most Significant Byte) and M3\_SLIN1\_01\_LSB (LSB for Less Significant Byte).

Select now M3\_SLIN1\_01\_LSB and click “Split” button (or press “s” on keyboard).

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_01_MSB	8	1
M3_SLIN1_01_B5-8	4	1
M3_SLIN1_01_B1-4	4	1

This time, the 8 bits data have been splitted into two 4 bits datas.

One more time, data names have changed according to their size and IO:

- M3\_SLIN1\_01\_B5-8 for data beginning at fifth bit and finishing at eighth bit of SL bloc output.
- M3\_SLIN1\_01\_B1-4 for data beginning at first bit and finishing at fourth bit of SL bloc output.

Continue to split each data beginning by “M3\_SLIN1\_01” in order to create one bit data per bit in SL bloc output as following.

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_01_B16	1	1
M3_SLIN1_01_B15	1	1
M3_SLIN1_01_B14	1	1
M3_SLIN1_01_B13	1	1
M3_SLIN1_01_B12	1	1
M3_SLIN1_01_B11	1	1
M3_SLIN1_01_B10	1	1
M3_SLIN1_01_B9	1	1
M3_SLIN1_01_B8	1	1
M3_SLIN1_01_B7	1	1
M3_SLIN1_01_B6	1	1
M3_SLIN1_01_B5	1	1
M3_SLIN1_01_B4	1	1
M3_SLIN1_01_B3	1	1
M3_SLIN1_01_B2	1	1
M3_SLIN1_01_B1	1	1
M3_SLIN1_02	16	2
M3_SLIN1_03	16	3
M3_SLIN1_04	16	4
M3_SLIN1_05	16	5
M3_SLIN1_06	16	6
M3_SLIN1_07	16	7
M3_SLIN1_08	16	8

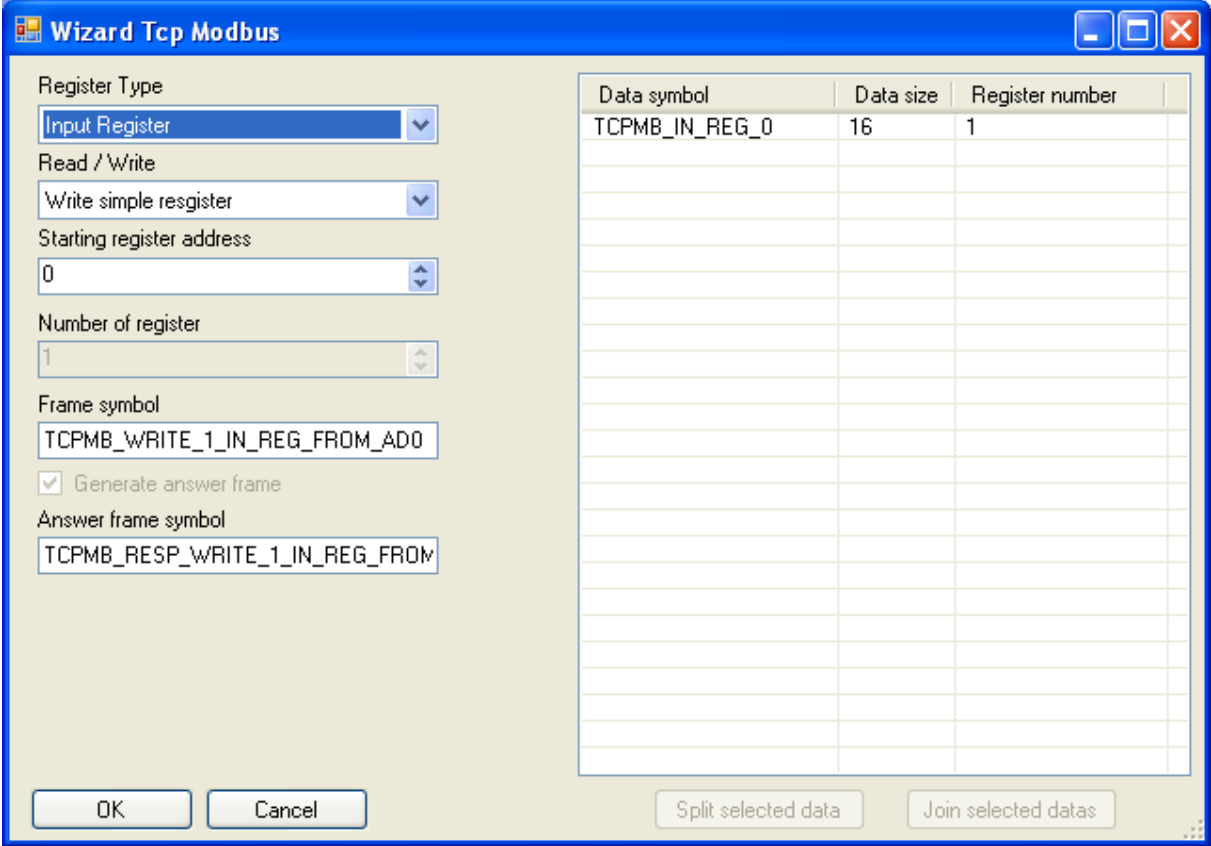
We have now sixteen 1 bit data for each bit in first SL Bloc output, and seven 16 bits datas for other SL Bloc outputs. Frame is now ready to be created.

Press “OK” button.

Before seeing what have been created, we will create writing frame for the SL input bloc 1-8. Open one more time the **M3 SL Bloc Wizard** and modify only the field “Read/Write” using “Write” choice. Split the first output data into sixteen 1 bit datas as done previously and click OK and then go to [Frame Window description](#).

### 2.2 TCP Modbus Wizard

Open TCP Modbus Wizard. The following window appears.



NOTE: the TCP modbus wizard have not been designed in order to be used only with Millenium 3 controllers but with any kind of controllers able to use this protocol.

- The left part of the window is used to configure frame for TCP Modbus communication:
- Register Type: select if you want to use an input or an output register. It is mainly used to create data names according to register type. Note that outputs registers can only be read.
  - Read/Write: select if you want to create a frame for reading or writing registers.
  - Starting register address: select the start address of read/written registers.
  - Number of registers: Only available for “Read holding registers” and “Write multiple registers” modbus orders.
  - Frame Symbol: Symbol of frame that will be created for sending read request on controller.
  - Answer frame Symbol: Symbol of frame that will be created to receive read request answer.

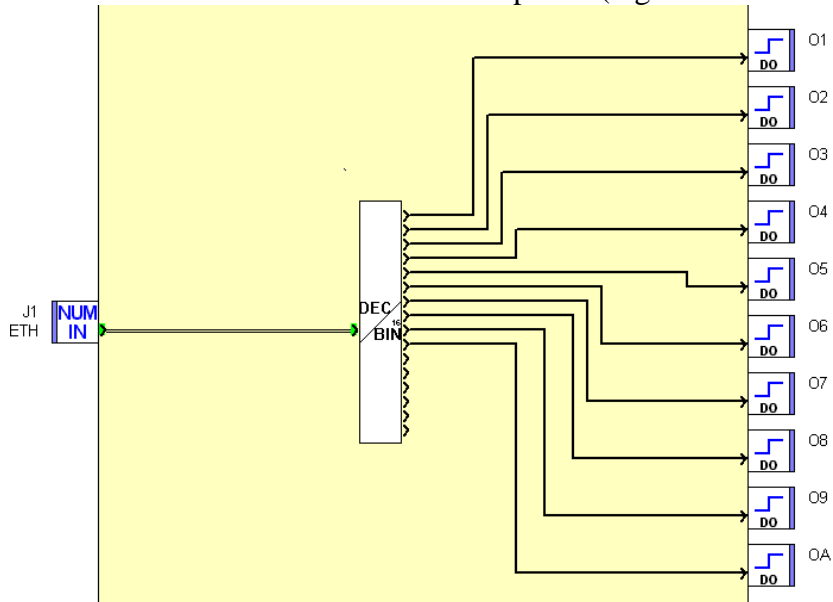
NOTE: M3 XN05 extension only support “Write simple register” and “Read Holding register” orders.



The right part of the window is the list of datas that will be used to store controller values. The list allows you to split each data to smaller size and at least create one data for each bit, quartet of byte for each register.

But see how it can be used concretely with **Ethernet Example (XN05).pm3** file

Here is a screen shot of M3 Ethernet Input J1 (register address 12) wiring in sample file.



The bloc output is connected to a DEC/BIN bloc that allows us to use each bit to order controller outputs. So we want to use each bit of NUM\_IN bloc output as independent data. Select wizard parameters as following

Register Type  
Input Register

Read / Write  
Read Holding register

Starting register address  
12

Number of register  
1

Frame symbol  
TCPMB\_READ\_1\_IN\_REG\_FROM\_AD12

Generate answer frame

Answer frame symbol  
TCPMB\_RESP\_READ\_1\_IN\_REG\_FROM\_

Click the first data named TCPMB\_IN\_REG\_12 and then click “Split selected data”.



Data symbol	Data size	Register number
TCPMB_IN_REG_12_B16	1	1
TCPMB_IN_REG_12_B15	1	1
TCPMB_IN_REG_12_B14	1	1
TCPMB_IN_REG_12_B13	1	1
TCPMB_IN_REG_12_B12	1	1
TCPMB_IN_REG_12_B11	1	1
TCPMB_IN_REG_12_B10	1	1
TCPMB_IN_REG_12_B9	1	1
TCPMB_IN_REG_12_B8	1	1
TCPMB_IN_REG_12_B7	1	1
TCPMB_IN_REG_12_B6	1	1
TCPMB_IN_REG_12_B5	1	1
TCPMB_IN_REG_12_B4	1	1
TCPMB_IN_REG_12_B3	1	1
TCPMB_IN_REG_12_B2	1	1
TCPMB_IN_REG_12_B1	1	1

We have now sixteen 1 bit data for each bit in register. Frame is now ready to be created. Press “OK” button.

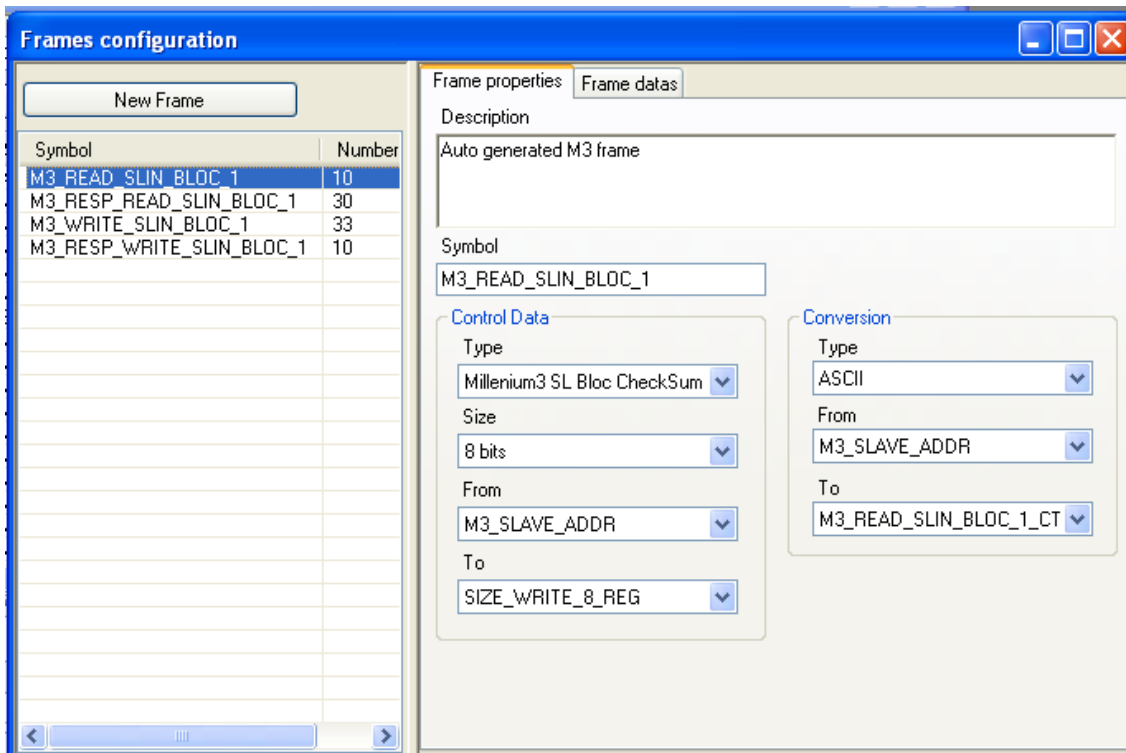
Before seeing what have been created, we will create writing frame for the SL input bloc 1-8. Open one more time the **TCP Modbus Wizard** and modify only the field “Read/Write” using “Write simple register” choice. Split the register data into sixteen 1 bit datas as done previously and click OK and then continue tutorial.

### 3. Frame Window

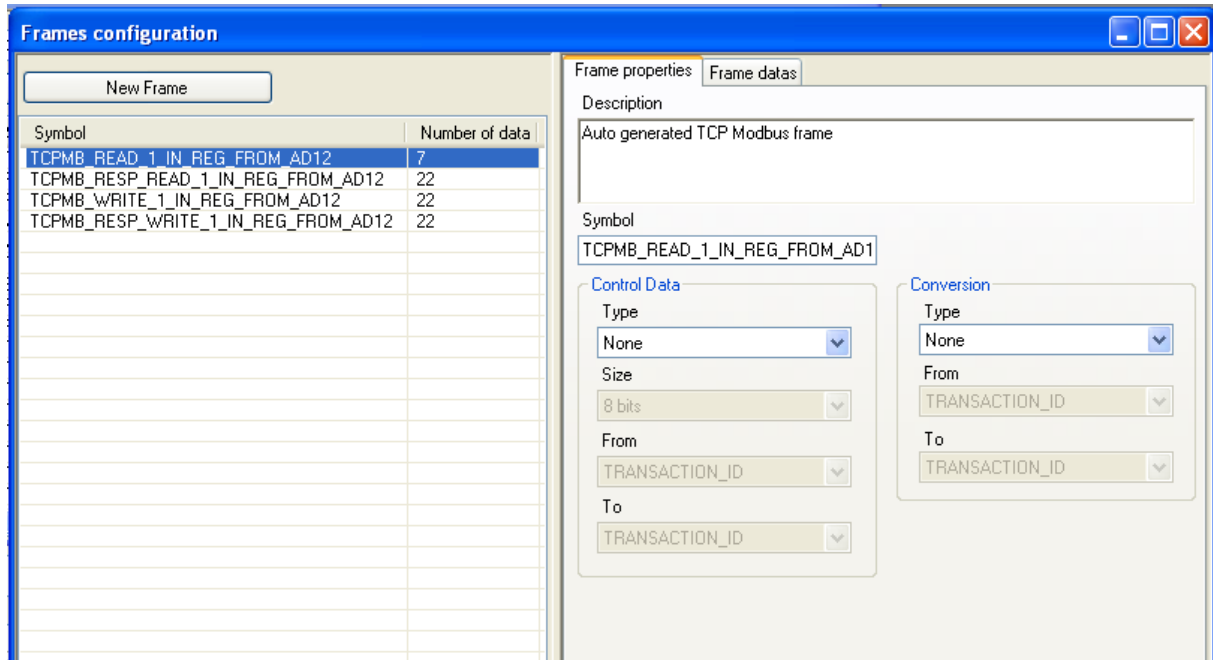
Activate “Frame Configuration” window in SmartCommand software.

According to witch wizard you have use, you must have one of following result:

M3 SL Bloc Wizard:



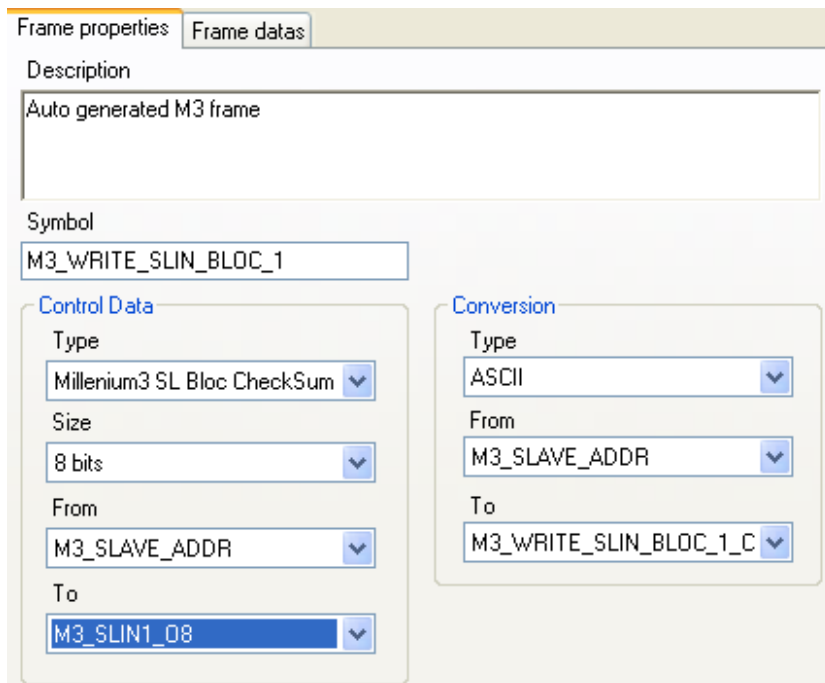
## TCP Modbus Wizard:



On the left side of window, you have the list of all existing frame, the one you have just created using wizards are here.

On the right part of the window, you have “Frame properties” and “Frame datas” pages.

### 3.1 Frame properties



On this page you can find followings parameters:

- Description: description of frame
- Symbol: symbol of frame
- Control Data group:

As many communications protocols need a checksum or a CRC, you can define witch kind of control data is used with your message. When you select a control data

different from none, a special data is automatically added to message, then you can move it to the right position. At the moment, the only allowed checksum is a M3 SL blocs checksum.

**NOTE:** if you want to use this software with controller witch need specific checksum or CRC, you can ask me to add it, provided that you gave me checksum or CRC algorithm and rules.

- Type: type of control data used in frame, available choices are none or M3SLBloc checksum.

- Size: size in bits of frame control data, it work as for standard data, but only 8, 16 and 32 bits sizes are authorized.

- From/To: some protocols build checksum on only a partial part of the frame, so here you can define the first and the last data that enter in checksum/CRC calculation. If selected data's have a size inferior to 1 byte, the checksum/CRC calculation begin from the byte where is "From data" in final frame, and end at the byte where is "To data" in final frame.

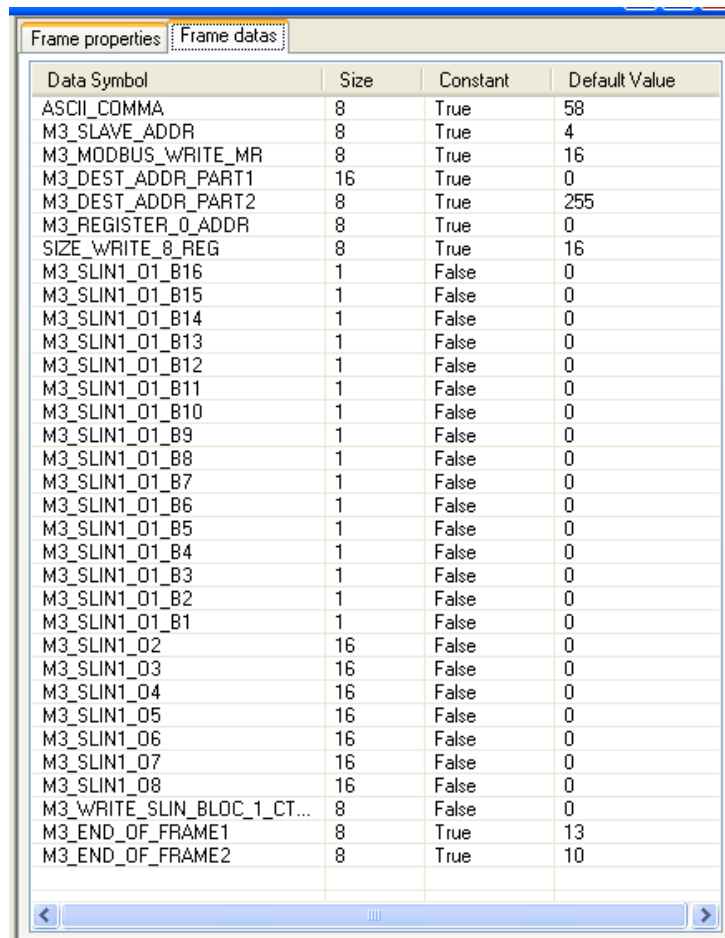
- Conversion Group:

Many communications protocols use a conversion. As for Controls Data, you can define witch type of conversion you want to use, the beginning and the end of conversion.

- Type: At the moment, the only available conversion is ASCII (used for M3 SL blocs communication).

- From/To: It works as for Control data

### 3.2 Frame datas



Data Symbol	Size	Constant	Default Value
ASCII_COMMA	8	True	58
M3_SLAVE_ADDR	8	True	4
M3_MODBUS_WRITE_MR	8	True	16
M3_DEST_ADDR_PART1	16	True	0
M3_DEST_ADDR_PART2	8	True	255
M3_REGISTER_0_ADDR	8	True	0
SIZE_WRITE_8_REG	8	True	16
M3_SLIN1_01_B16	1	False	0
M3_SLIN1_01_B15	1	False	0
M3_SLIN1_01_B14	1	False	0
M3_SLIN1_01_B13	1	False	0
M3_SLIN1_01_B12	1	False	0
M3_SLIN1_01_B11	1	False	0
M3_SLIN1_01_B10	1	False	0
M3_SLIN1_01_B9	1	False	0
M3_SLIN1_01_B8	1	False	0
M3_SLIN1_01_B7	1	False	0
M3_SLIN1_01_B6	1	False	0
M3_SLIN1_01_B5	1	False	0
M3_SLIN1_01_B4	1	False	0
M3_SLIN1_01_B3	1	False	0
M3_SLIN1_01_B2	1	False	0
M3_SLIN1_01_B1	1	False	0
M3_SLIN1_02	16	False	0
M3_SLIN1_03	16	False	0
M3_SLIN1_04	16	False	0
M3_SLIN1_05	16	False	0
M3_SLIN1_06	16	False	0
M3_SLIN1_07	16	False	0
M3_SLIN1_08	16	False	0
M3_WRITE_SLIN_BLOC_1_CT...	8	False	0
M3_END_OF_FRAME1	8	True	13
M3_END_OF_FRAME2	8	True	10

On this page you can find data list of currently selected frame. Here you can modify order using drag and drop in list.

**WARNING:** You should not modify frame which have been created by wizard since they have been tested and configured specially for M3 SL bloc or TCP modbus protocol. You can do it only if you know precisely how build protocols are.

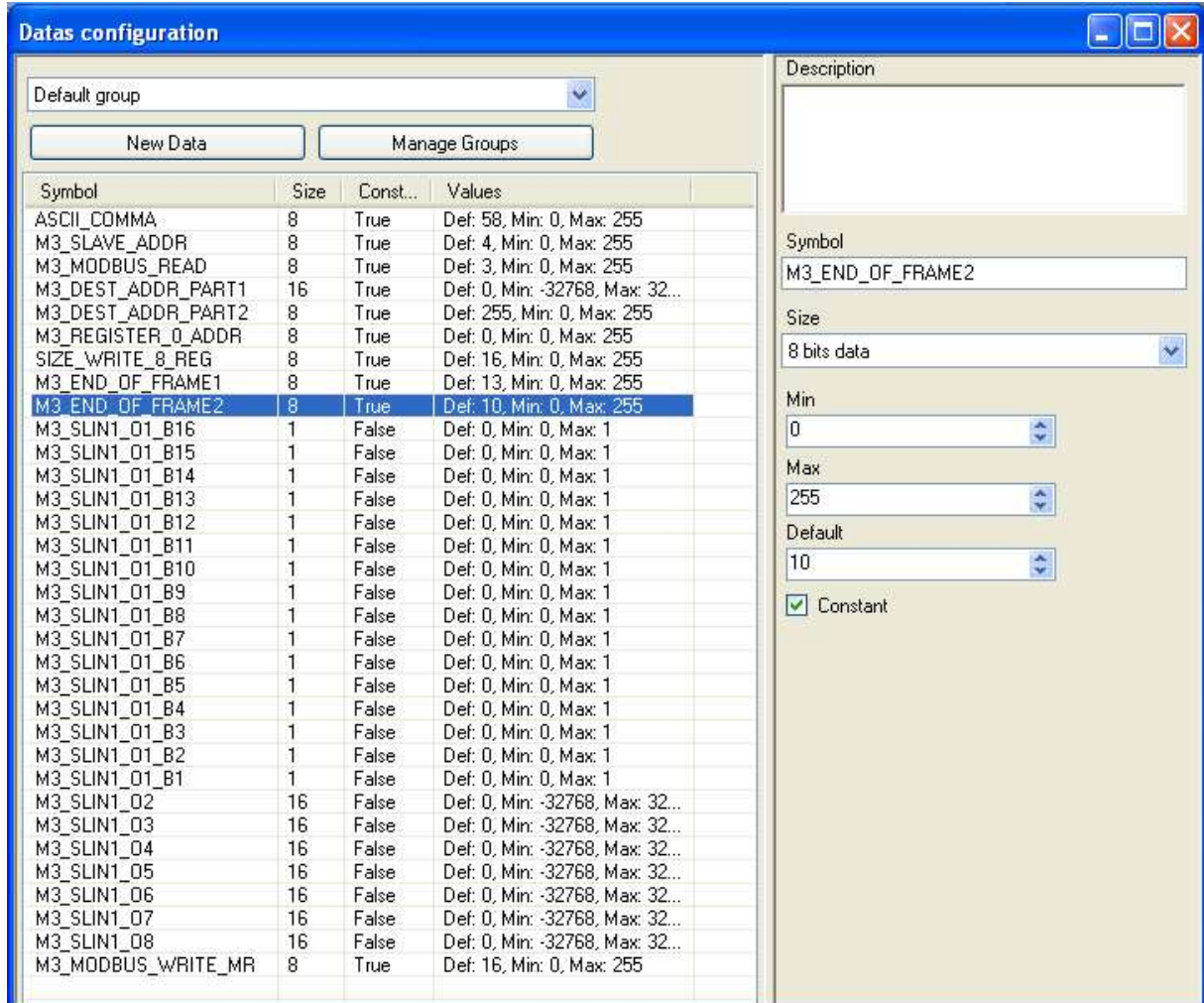
## 4. Data Window

Activate “Data Configuration” window in SmartCommand software.

When the frame have been created using wizards, all datas used by frames have been created.

According to witch wizard you have use, you must have one of following result:

M3 SL Bloc Wizard:



The screenshot shows the 'Datas configuration' window. The left pane contains a table of data symbols. The right pane shows the configuration for the selected symbol, 'M3\_END\_OF\_FRAME2'.

Symbol	Size	Const...	Values
ASCII_COMMA	8	True	Def: 58, Min: 0, Max: 255
M3_SLAVE_ADDR	8	True	Def: 4, Min: 0, Max: 255
M3_MODBUS_READ	8	True	Def: 3, Min: 0, Max: 255
M3_DEST_ADDR_PART1	16	True	Def: 0, Min: -32768, Max: 32...
M3_DEST_ADDR_PART2	8	True	Def: 255, Min: 0, Max: 255
M3_REGISTER_0_ADDR	8	True	Def: 0, Min: 0, Max: 255
SIZE_WRITE_8_REG	8	True	Def: 16, Min: 0, Max: 255
M3_END_OF_FRAME1	8	True	Def: 13, Min: 0, Max: 255
<b>M3_END_OF_FRAME2</b>	<b>8</b>	<b>True</b>	<b>Def: 10, Min: 0, Max: 255</b>
M3_SLIN1_01_B16	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B15	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B14	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B13	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B12	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B11	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B10	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B9	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B8	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B7	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B6	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B5	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B4	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B3	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B2	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_01_B1	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_02	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_03	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_04	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_05	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_06	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_07	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_08	16	False	Def: 0, Min: -32768, Max: 32...
M3_MODBUS_WRITE_MR	8	True	Def: 16, Min: 0, Max: 255

The right-hand pane shows the configuration for 'M3\_END\_OF\_FRAME2':

- Symbol: M3\_END\_OF\_FRAME2
- Size: 8 bits data
- Min: 0
- Max: 255
- Default: 10
- Constant

## TCP Modbus Wizard:

The screenshot shows the 'Datat configuration' window. The left pane contains a table with the following data:

Symbol	Size	Const...	Values
TRANSACTION_ID	16	True	Def: 1, Min: -32768, Max: 32...
PROTOCO_ID	16	True	Def: 0, Min: -32768, Max: 32...
FOLLOWING_6_BYTES	16	True	Def: 6, Min: -32768, Max: 32...
MODBUS_SLAVE_ADDR	8	True	Def: 1, Min: 0, Max: 255
MODBUS_READ_REG	8	True	Def: 3, Min: 0, Max: 255
TCPMB_REG_ADDR_12	16	True	Def: 12, Min: -32768, Max: 3...
BYTE_READ_1_REG	16	True	Def: 1, Min: -32768, Max: 32...
FOLLOWING_5_BYTES	16	True	Def: 5, Min: -32768, Max: 32...
READ_RECIEVE_2_BYTES	8	True	Def: 2, Min: 0, Max: 255
TCPMB_IN_REG_12_B16	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B15	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B14	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B13	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B12	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B11	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B10	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B9	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B8	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B7	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B6	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B5	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B4	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B3	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B2	1	False	Def: 0, Min: 0, Max: 1
TCPMB_IN_REG_12_B1	1	False	Def: 0, Min: 0, Max: 1
MODBUS_WRITE_SR	8	True	Def: 6, Min: 0, Max: 255

The right pane shows the 'Description' field, a 'Symbol' field, and a 'Size' dropdown menu set to '8 bits data'. Below these are 'Min' and 'Max' dropdown menus, both set to '0', and a 'Default' dropdown menu also set to '0'. A 'Constant' checkbox is present and unchecked.

The left part of this window contains the list of all datas according to currently selected group. Using the “Manage Groups” button, you can open a new window allowing you create new groups and then organize you data according to their purpose. I will not detail this part as it has no functional interest.

The data list displays all information’s about each data. Each parameter will be detailed with the right part of the window:



Description

Symbol  
TCPMB\_IN\_REG\_12\_B16

Size  
1 bit data

Min  
0

Max  
1

Default  
0

Constant

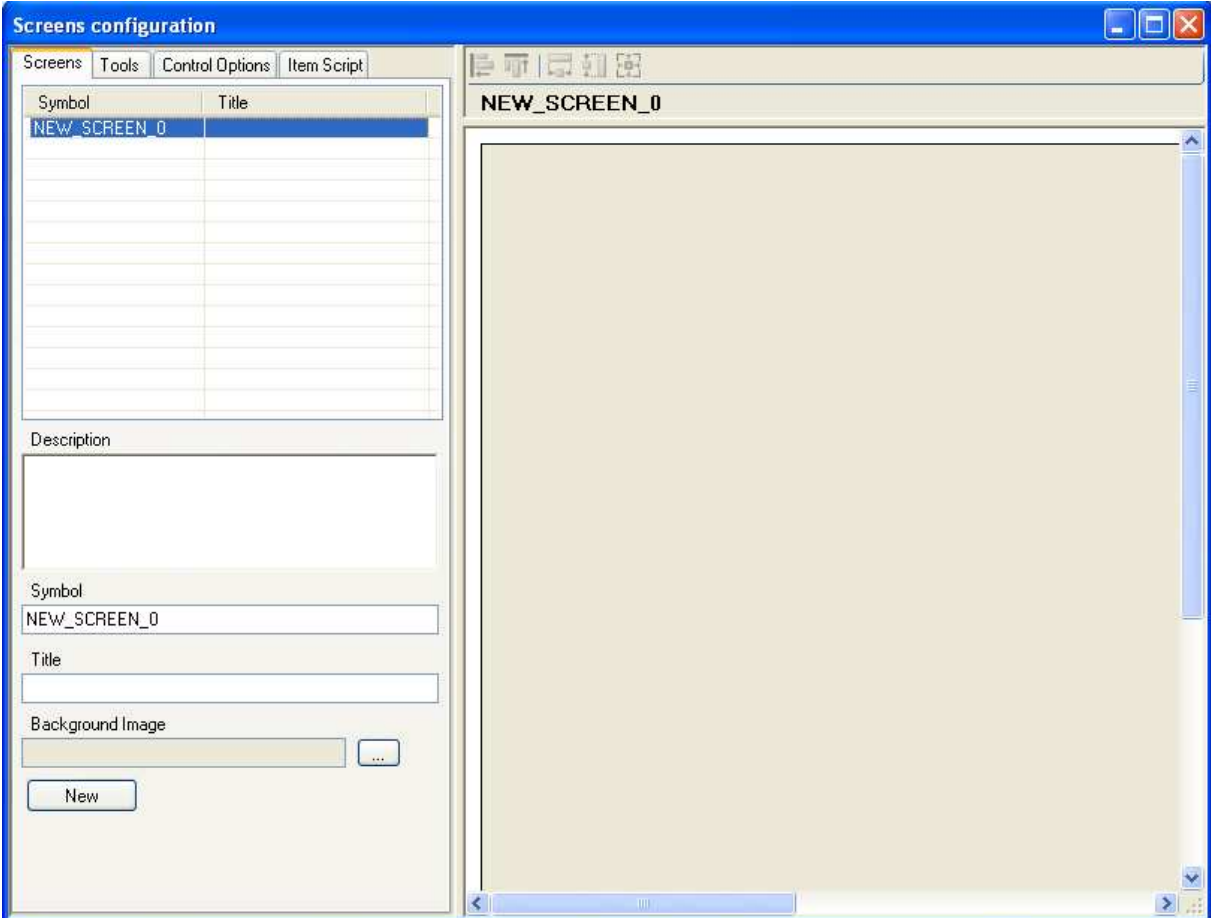
On the right part of window you can find followings parameters:

- Description: description of data
- Symbol: symbol of data
- Size: Size if data in bits
- Max: maximum value that can have the data. Range change according to data size.
- Min: minimum value that can have the data. Range change according to data size.
- Default: Default value of data when used (see SmartCommand)
- Constant: if this option is checked, the value will be used as a constant and will always have its default value.

NOTE: "Constant" parameter may be critical as it is used on frame reception. Each constant data are compared to receive value (and must be equal), unlike non constant datas witch are assigned to receive value.

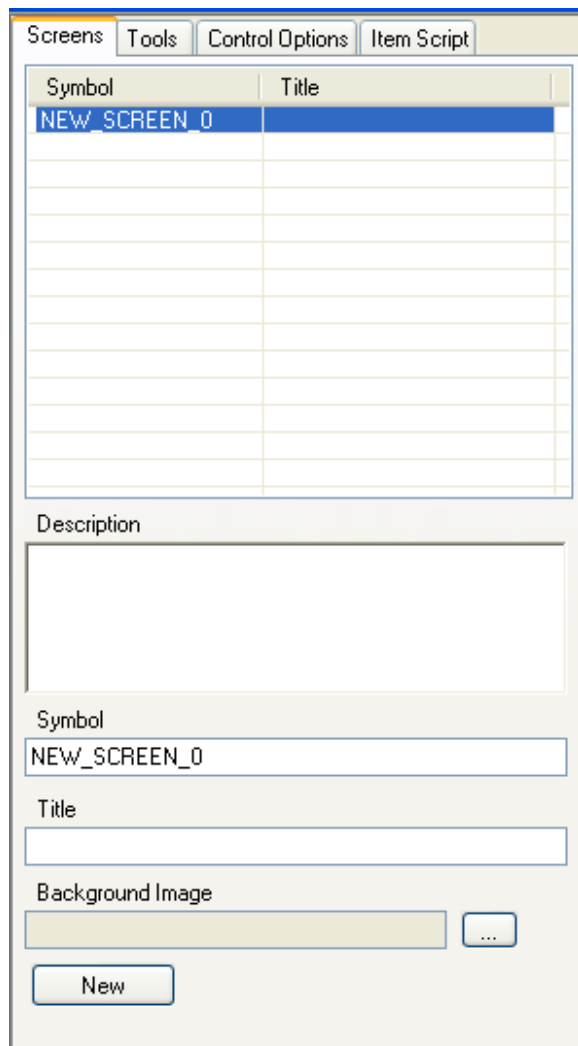
There's nothing more to do in this window since all data should not be modified.

# 5. Screen Window



In this window, you can create and design appearance and behavior of displayed screens in SmartCommand software.

## 5.1 Screen Page



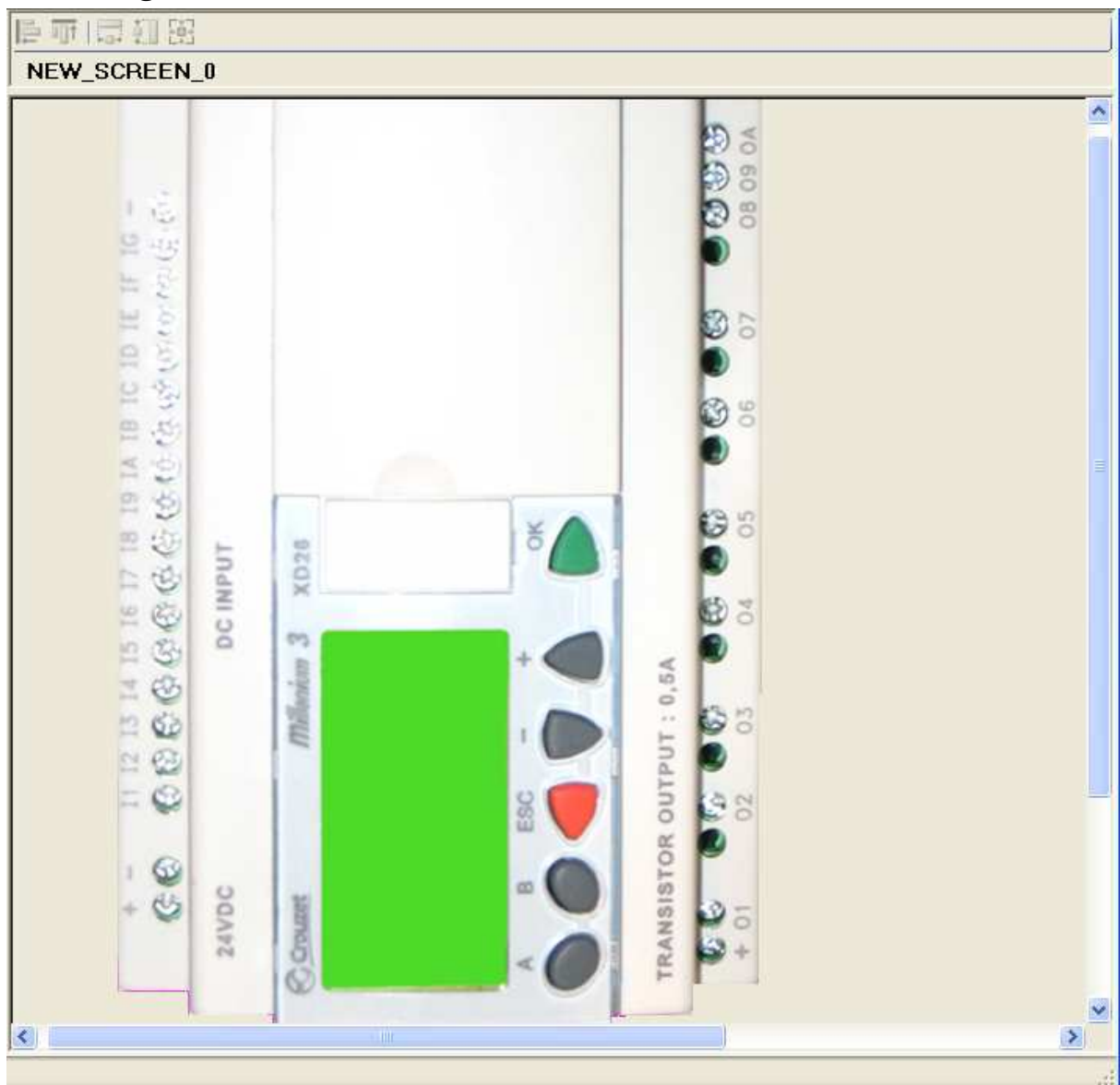
On the top part, you can find a list of all existing screens. Press the “New” button and then select the new screen in top list.

On bottom part you can find screen parameters.

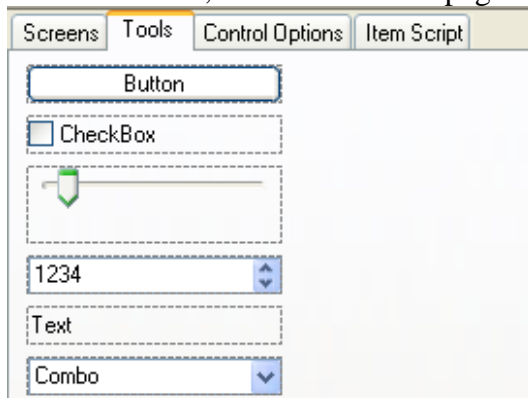
- Description: description of screen
- Symbol: symbol of screen
- Title: Title that will have screen in SmartCommand application
- Background image: path to an image file that will be used as a background for screen.

Enter the Title “Millenium 3 Controller”, and then press the “...” button to browse an image file. In Application directory you will find usable image of a Millenium 3 controller. The path to application folder should be: System Disk letter... most often “C”\program Files\M3Tool\SmartApp\Examples. Select the file **M3XD26S.bmp** and then click in right part of the window to validate changes. The selected picture is now displayed in “Designer”.

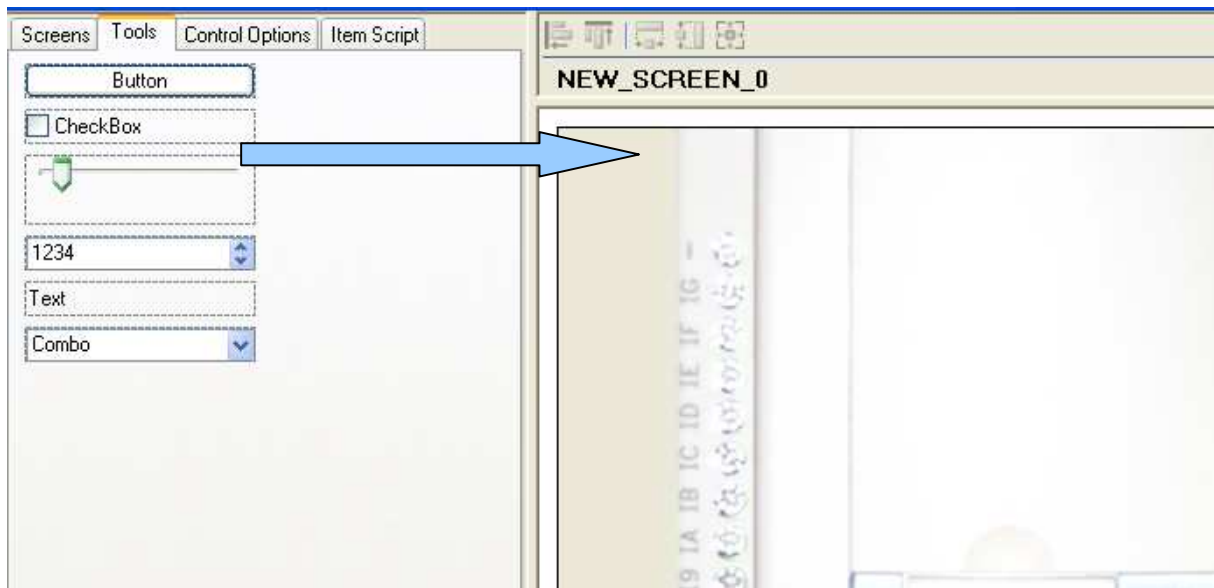
## 5.2 Designer



Here is the designer with the newly selected background image. On the left side, select the “Tool” page.



In this page you can find all available kind of controls that can be used in screens. You can drag and drop any of them in designer area.



Once a control has been added, you can move and resize it in the designer area by selecting the item, and then using the move button or selection squares.



For current purposes, we will not use button controls, so select the one that you have dropped and press the "Suppr" key to delete the button.

While in frame configuration, we have built a frame for reading and writing a register connected to controller outputs. So place a "CheckBox" control for each controller output in the designer as following and select them all (using mouse square selection or shift and mouse click).



Now click the “Arrange left” button



All Item will be aligned on left. If you want you can also move them all using “move” button



Now, double click on the item in front of controller output 1. The “Control Option” page is displayed.

### **5.3 Control Options Page**

On bottom part you can find screen parameters.

- Description: description of screen
- Symbol: symbol of screen

On bottom part you can find screen parameters.

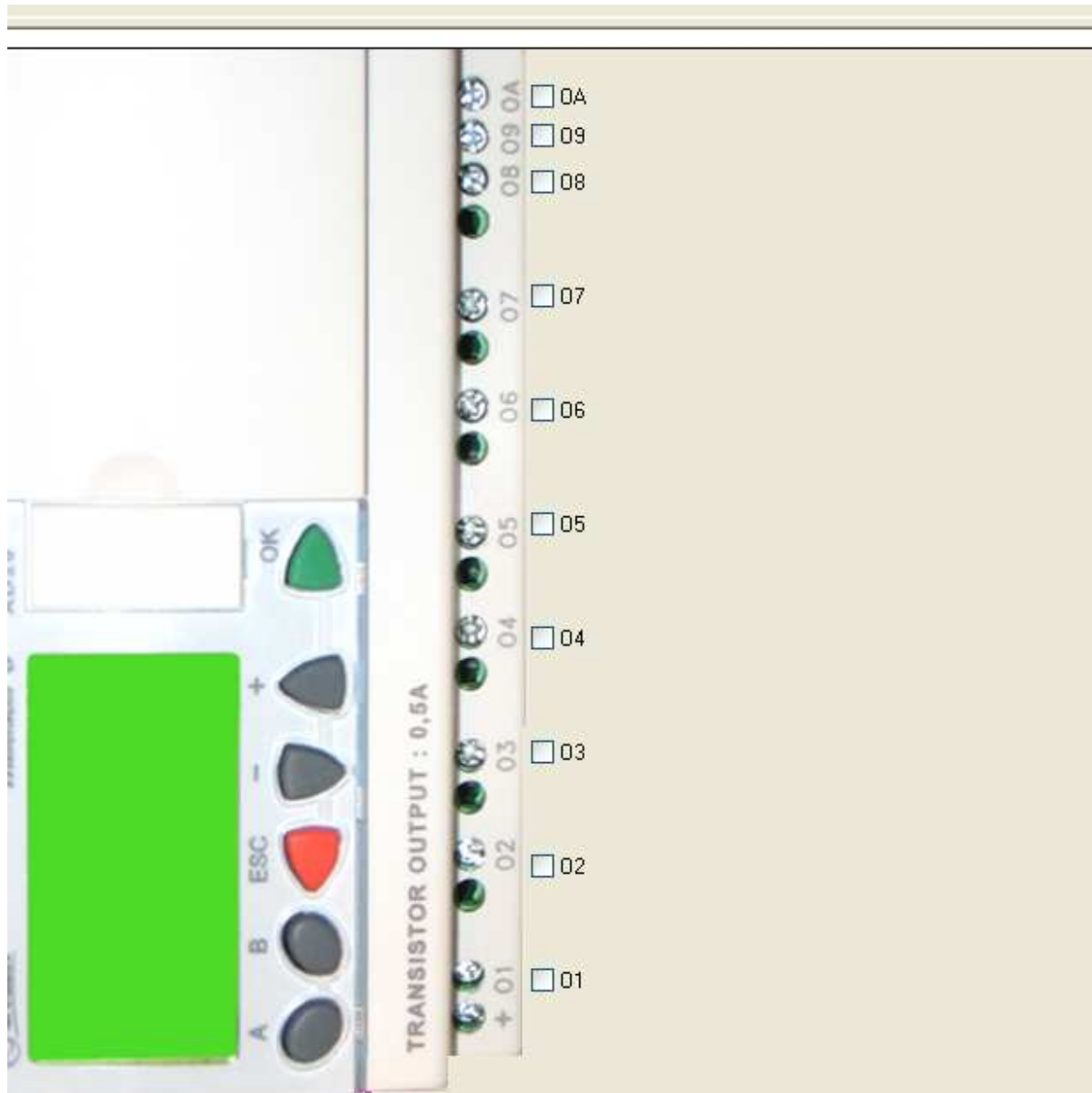
- Description: description of screen
- Symbol: symbol of screen

-

On this page you can find control parameters.

- Description: description of control
- Symbol: symbol of control
- Associate Data: Symbol of a data that will be used by control in Smart Command software. This property is one of the most important as it will decide the behavior of control according to data value.
- Text: The text displayed by the control
- Read Only: set if the control is a read only one (value could not be modified by human action)
- Use Screen Event: determine if the control raised the screen event and so, execute the script (see following chapter for more informations about script).

At this step you should change text according to controller outputs and active “Use screen event” properties of each control like in following screen shots. (Note, change made to a control are applied when you leave the control properties page, by clicking anywhere outside for example.



## Associating datas and controls

This part is very important as it will set which values exchanged with remote controller will be displayed.

In our purpose, we would like each check box controlling one output of remote controller. When frames have been created, datas have been created for each bits of remote controller register connected to controllers outputs through a DEC/BIN bloc. So each data will correspond to controller outputs.

TCP modbus Wizard:

TCPMB\_IN\_REG\_12\_B1: Controller O1

TCPMB\_IN\_REG\_12\_B2: Controller O2

TCPMB\_IN\_REG\_12\_B3: Controller O3

TCPMB\_IN\_REG\_12\_B4: Controller O4

TCPMB\_IN\_REG\_12\_B5: Controller O5

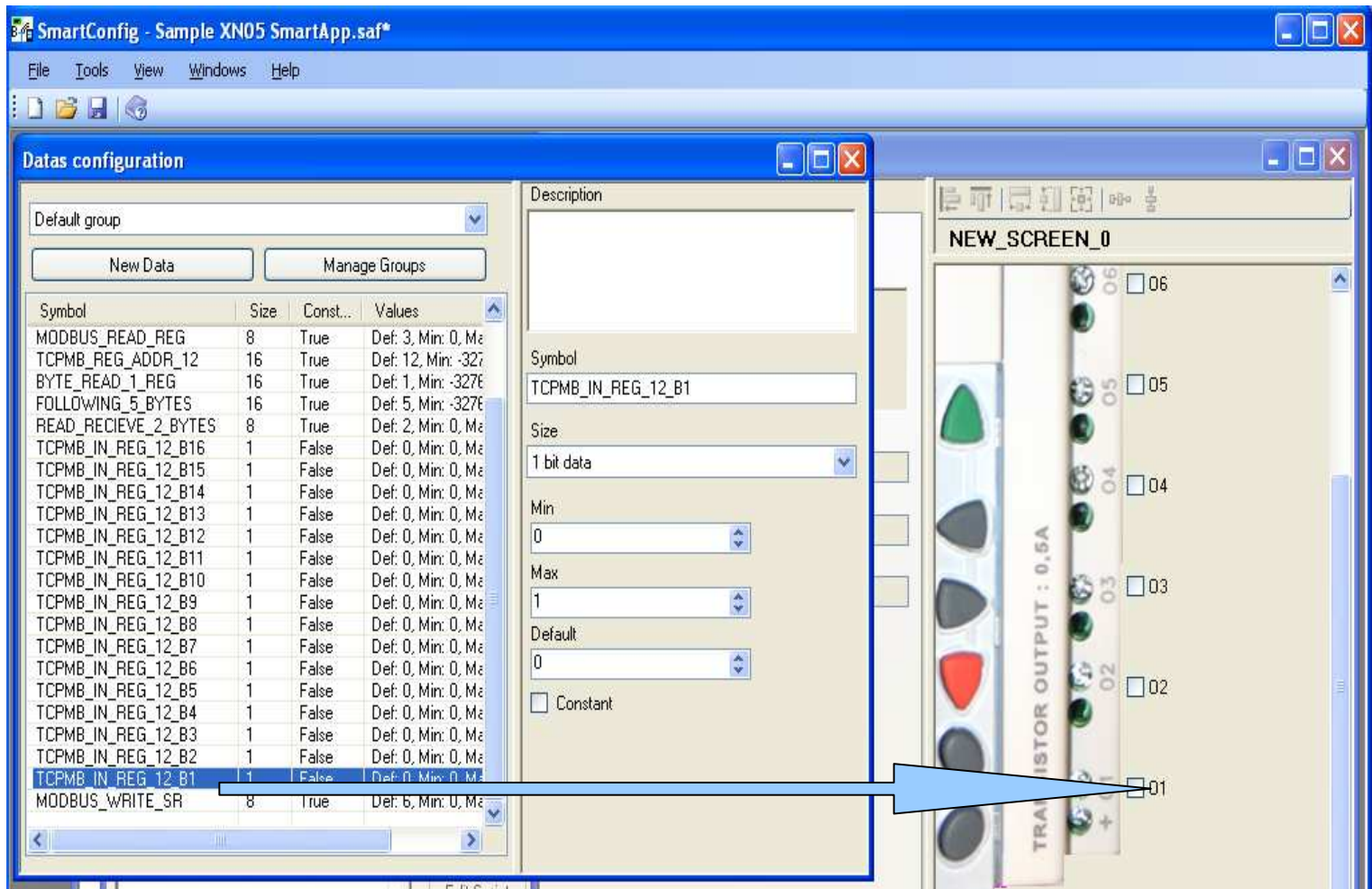


TCPMB\_IN\_REG\_12\_B6: Controller O6  
 TCPMB\_IN\_REG\_12\_B7: Controller O7  
 TCPMB\_IN\_REG\_12\_B8: Controller O8  
 TCPMB\_IN\_REG\_12\_B9: Controller O9  
 TCPMB\_IN\_REG\_12\_B10: Controller OA

M3 SL Bloc Wizard:

M3\_SLIN\_O1\_B1: Controller O1  
 M3\_SLIN\_O1\_B2: Controller O2  
 M3\_SLIN\_O1\_B3: Controller O3  
 M3\_SLIN\_O1\_B4: Controller O4  
 M3\_SLIN\_O1\_B5: Controller O5  
 M3\_SLIN\_O1\_B6: Controller O6  
 M3\_SLIN\_O1\_B7: Controller O7  
 M3\_SLIN\_O1\_B8: Controller O8  
 M3\_SLIN\_O1\_B9: Controller O9  
 M3\_SLIN\_O1\_B10: Controller OA

Arrange “Data” and “Screen” windows as in following screen shot.  
 Select and drag the data named TCPMB\_IN\_REG\_12\_B1 (for TCP modbus) or  
 M3\_SLIN1\_O1\_B1 over the check box named “O1”



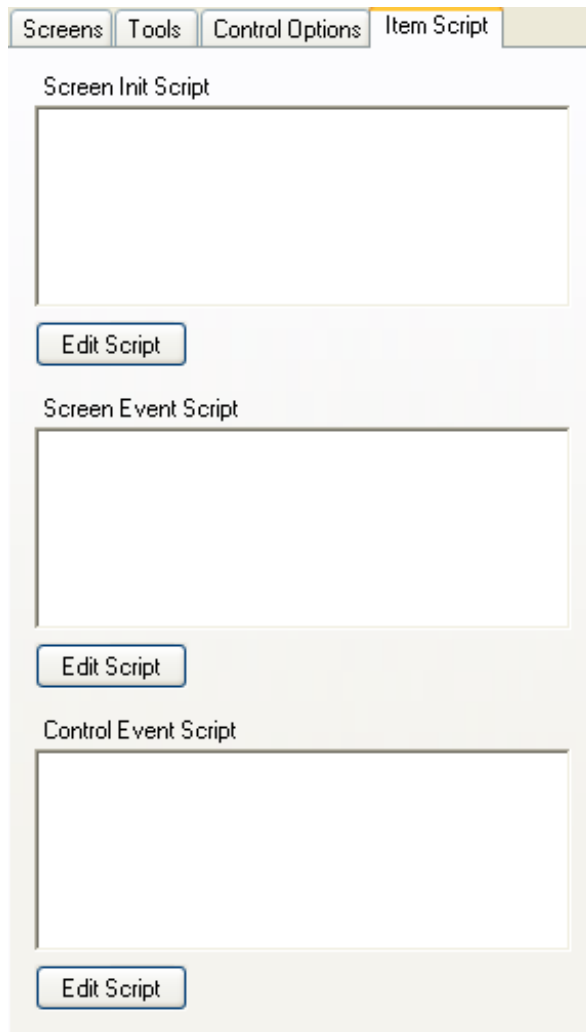
Once the mouse cursor hover the check box, drop become available. Release mouse button. A green light appear on top left of check box, data have been associated. Now value displayed by control will depend of data value, and data value will be modified by actions on control in Smart Command software.

Repeat this operation for each controller outputs with correspondence table given previously.

Save your file and go to next step.

## 5.4 Item Script Page

Active now the Item Script Page.



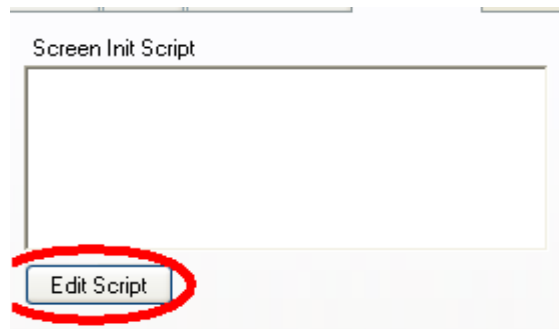
The screenshot displays a software interface with a tabbed menu at the top containing 'Screens', 'Tools', 'Control Options', and 'Item Script'. The 'Item Script' tab is selected. Below the menu, there are three distinct sections for script editing:

- Screen Init Script:** A large empty text area with an 'Edit Script' button below it.
- Screen Event Script:** A large empty text area with an 'Edit Script' button below it.
- Control Event Script:** A large empty text area with an 'Edit Script' button below it.

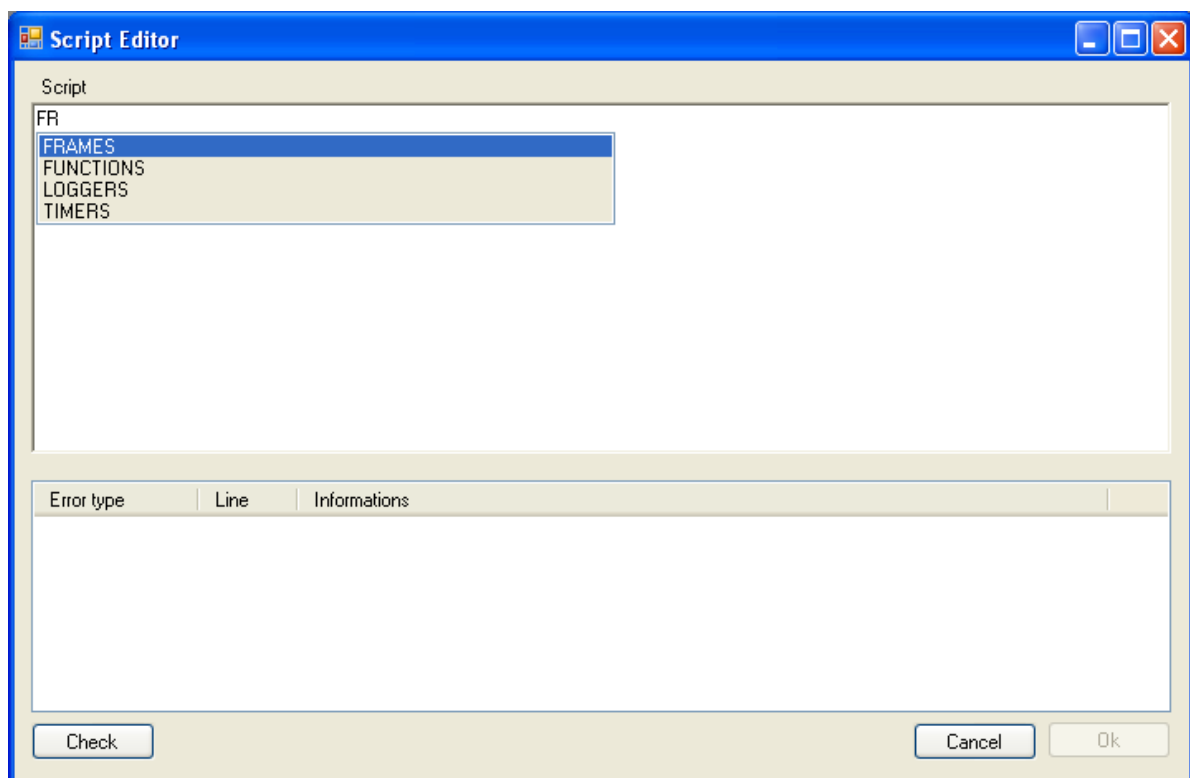
In this page, you can find 3 text box containing that will be executed by screen or control at different moment:

“Screen Init Script” : this script will be executed at startup of configuration file when used in Smart Command software. In this script, you should add all initialization that you will need for your application. In this tutorial we will need to initialize our data with values in remote controller. During at the beginning of the tutorial, we have used wizards and created frame for reading and writing remote registers.

Press the “Edit Script” button for “Init Script”.



## 6. Script



This window will help you to write script instructions using a contextual help giving you all available choices.

The top part of the window contain the script, and the bottom contains error window that you will have to used to verify your script consistency.

For the moment, script have not many abilities but in spite of this weakness, it is very easy to understand.

Most of time, a line can be decomposed in 3 elements separated by points.

- 1: the object type
- 2: the object symbol
- 3: and the object instruction

See an example with this line:

```
FRAMES . TCPMB_READ_1_IN_REG_FROM_AD12 . SEND()
```

In this line, the first element is “FRAMES”. It means that the line will execute a frame instruction.

The second element is the frame symbol “TCPMB\_READ\_1\_IN\_REG\_FROM\_AD12”. It means that the instruction will be executed by the specified frame. (here, the frame have been generated using TCP/Modbus wizard.)

The last element is the instruction to execute followed by brackets. “SEND()”.

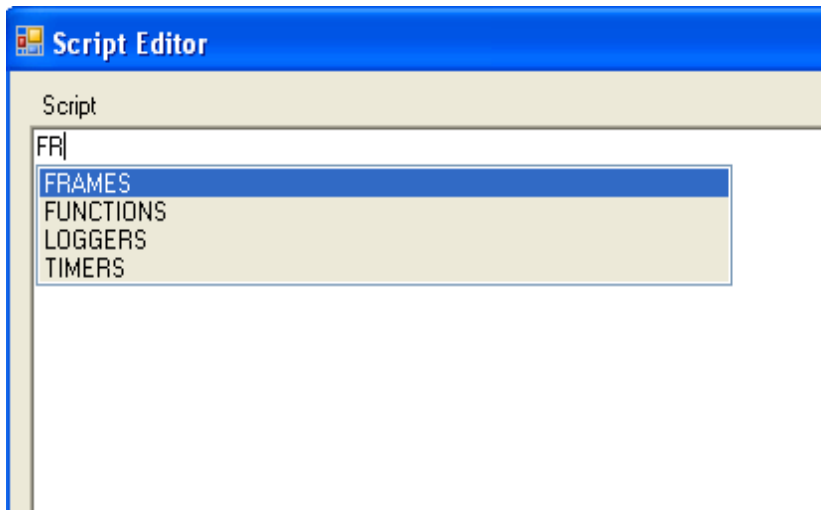
The whole line means that frame TCPMB\_READ\_1\_IN\_REG\_FROM\_AD12 will be send to connected remote controller on script execution.

### **Object Type:**

- Frames:
  - Keyword: FRAMES
  - Description: Frames are “objects” exchanged with remote controller to read or write data into its registers.
- Functions:
  - Keyword: FUNCTIONS
  - Description: functions are objects containing only script and can be executed in others scripts. Functions should be used create a script that will be used as only one line in any other script. For example, you can create a function that send and receive many frames and used it in screens for all items executing the same action instead of re-write all the script for each items
- Loggers:
  - Keyword: LOGGERS
  - Description:Logger allow you to store data values in “.csv” files. These files can be used in other softwares like Excel© to extract values as graphics. More details about logger are given here.
- Timers
  - Keyword: TIMERS
  - Description: Timer are used to execute periodically a script. The most interesting use of these object is to execute periodically read instructions on remote controller in order to supervise specifics registers values (according to sent frames). More details about Timers are given here

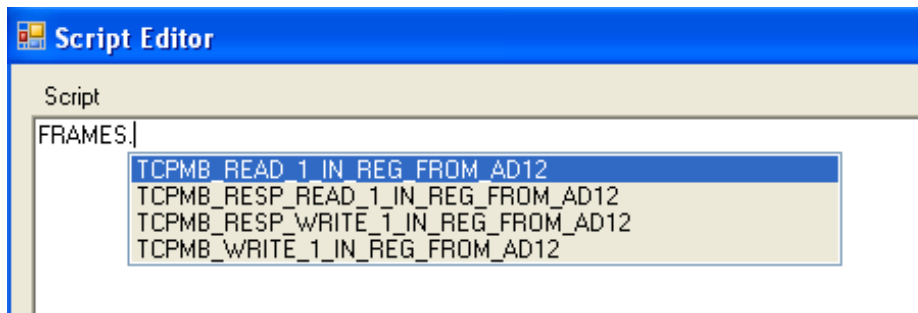
### **Back to tutorial:**

start writing “FRAMES” keyword in script editor. A List of choice appear.



In this list you can find all kind of object available as see previously. By pressing the “Down Arrow” key on your keyboard, you will active the “Auto completion List”. Once the list is activated, you can change selected value using up and down arrow keys. Select FRAMES and then press enter.

The text “FRAMES” have been written in script editor. Now, add a point. On more time the list appear but with other choices.



Here is the list of all frames crated using wizard (here is for TCP Modbus wizard). Whether for SL Bloc protocol or for TCP modbus protocol. A full exchanged between software an controller contains a request (read or write) and an answer corresponding to request. In our case we would like to read remote controller registers for initializations. So we will send one read frame request and receive one answer frame.

In script it would be written like that:

we first send the read request to controller

**FRAMES.TCPMB\_READ\_1\_IN\_REG\_FROM\_AD12.SEND()**

and then we wait and read controller answer.

**FRAMES.TCPMB\_RESP\_READ\_1\_IN\_REG\_FROM\_AD12.RECEIVE()**

For M3 SL Bloc wizard, the script would be:

**FRAMES.M3\_READ\_SLIN\_BLOC\_1.SEND()**

**FRAMES.M3\_RESP\_READ\_SLIN\_BLOC\_1.RECEIVE()**

Once the script is written, click “Check” button in bottom left, is there is no errors, the “OK” button become available.

Press OK button and then open the “Screen Event Script”.

This script will be executed when clicking in any check boxes that you have added to current screen (if “Use screen Event” attribute is set), and we want check boxes executing frames sens to write controller registers.

Using writing frames works in the same way as reading, we first send a write request, and then read the answer.

The script with TCP modbus wizard would be :

```
FRAMES.TCPMB_WRITE_1_IN_REG_FROM_AD12.SEND()  
FRAMES.TCPMB_RESP_WRITE_1_IN_REG_FROM_AD12.RECEIVE()
```

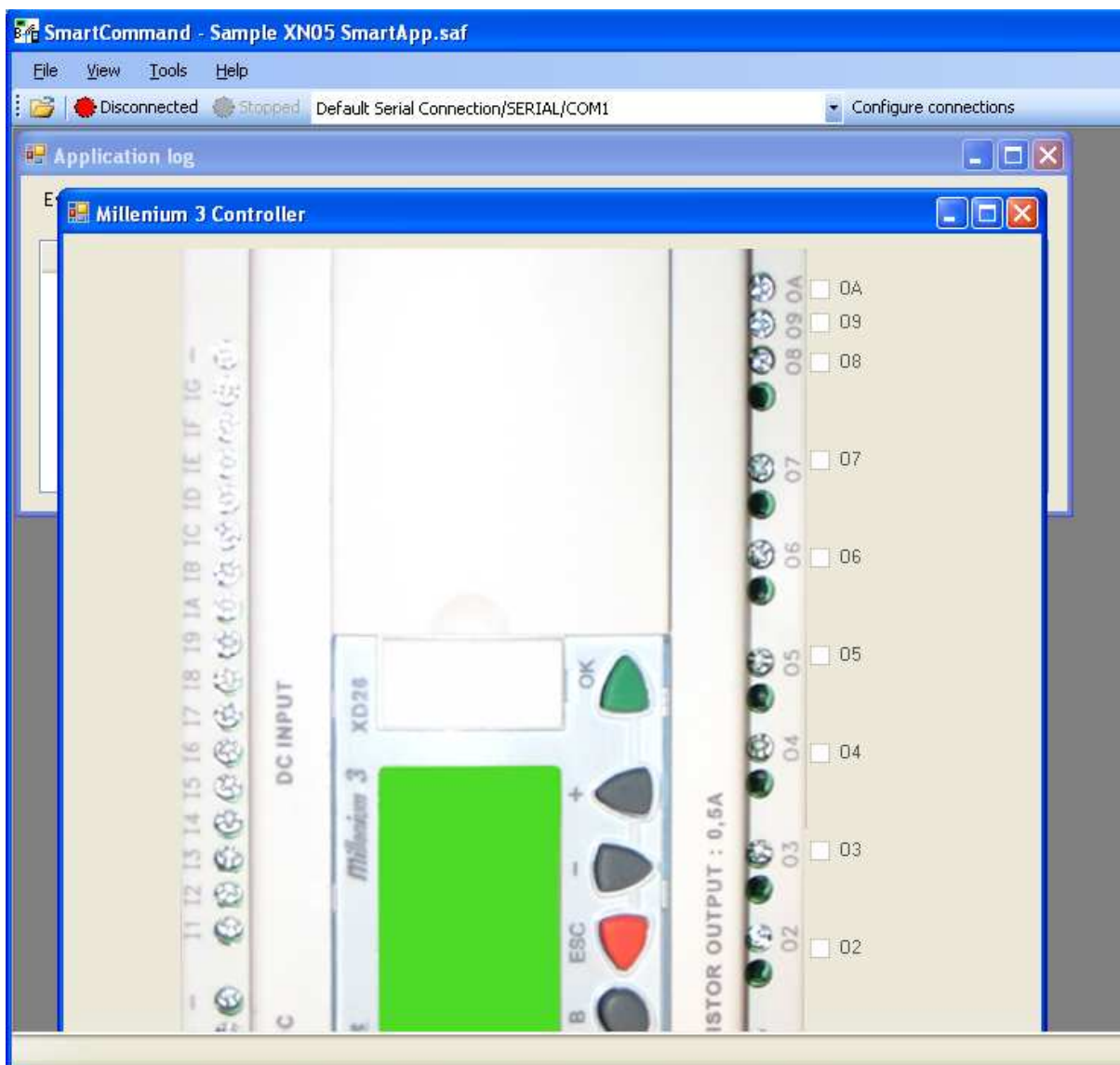
For ML3 SL Bloc Wizard:

```
FRAMES.M3_WRITE_SLIN_BLOC_1.SEND()  
FRAMES.M3_RESP_WRITE_SLIN_BLOC_1.RECEIVE()
```

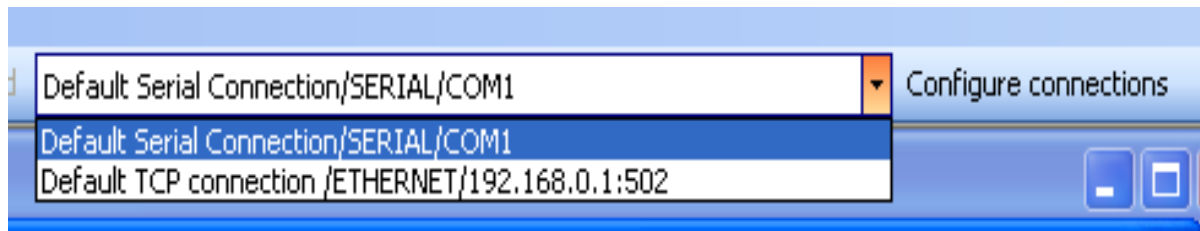
**Once scripts are written, save your file and use tool menu and then “Jup To Smart Command” (or F5 key) the program hides and then smart command launch and load automatically current file.**

## 7. SmartCommand

The purpose of this software is to use created file to supervise remote controllers. The Smart Command software appear as following.

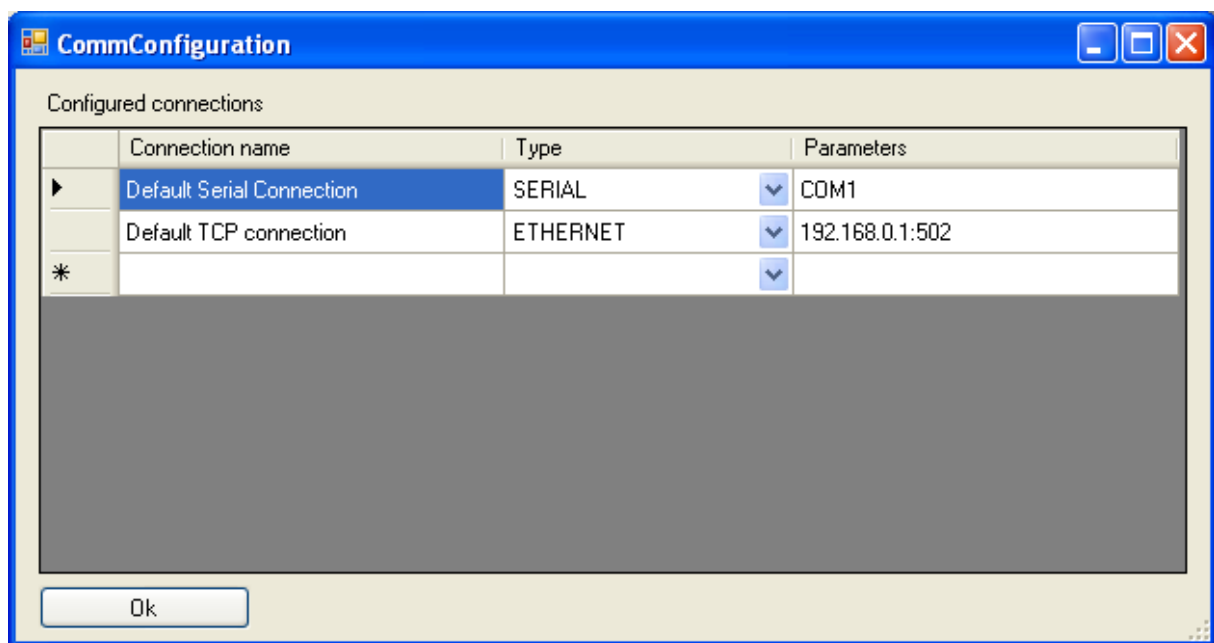


## 7.1 Connection configuration



In tool bar, you can find connection drop list, and connections configuration button. By default, two different connection are configured.(see screen shot)

By pressing on “Configure connections” button, following window appear:



In order to use the file with your remote controller, you should create a new or modify existing connections.

In the first column, you can enter a connection name, in second column, you must select connection type: Ethernet or Serial (use Serial for Bluetooth). In last column, you must enter connection parameters:

For Serial: the COM port (COMn where n is the number of valid serial)

For Ethernet: IP address of your remote controller , colon, TCP port number (502 for TCP Modbus Protocol).

Once you've added your connections, press OK button.

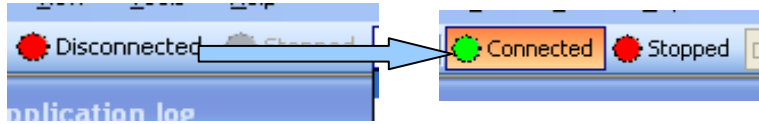
Note: Configured connection are automatically saved and are restored on next software startup.

The last connection used with a configuration file is saved on software exit and will be restored on next file opening. This behavior can be modified using option page.

## 7.2 Running program

Select your own connection in connection's drop list, and then press “Disconnected” button.





Once software is connected to remote controller, the button state changed, and the “Stopped” button becomes available.

Since the current loaded file is not started, nothing happens. When you press the “Stopped” button, it's state change and software execute initializations written in “Screens Init Script” If automatic objects are configured like Timers and Loggers, they are also started (theses objects will be describe later).

Screen Item are now available. Since the controller have been started, all registry value should be null and so, all check boxes should be unchecked.

Click of any of them. The value is written on controller register by executing “Screen Event Script” if you use an non static relay controller you should have heard controller relays closing corresponding to output check box, if you use a static relay controller (such as XD10S or XD26S) you can see the outputs states on controller front panel. Test that each check box actuate corresponding controller output.

### Summary:

In this first part, you have learn how to create frames, screens and configure them to read/write remote controller registers.

Theses actions can be executed periodically by software, and so it become able to refresh its state according to controller register values. See it in next chapter.

Stop your configuration and then

## 8.Finalize M3 Controller Supervision

Close Smart Command Software, “Smart Configuration” software reappear.

When supervising a remoter controller, the most interesting is reading values of controller outputs registers. As we have not created needed frames, active the wizards according to your current project type (SL Bloc or TCP Modbus). You can refer to Wizards section for help.

The frames parameters should be:

### M3 SL Bloc:

Frame Type = M3 SL Output Bloc

Read/Write = Read (can't be modified)

Address range = 25-32

M3\_SLOUT1\_I1 register splitted into 16 one bit value. (as in wizard tutorial)

### TCP Modbus:

Register Type = Output Register

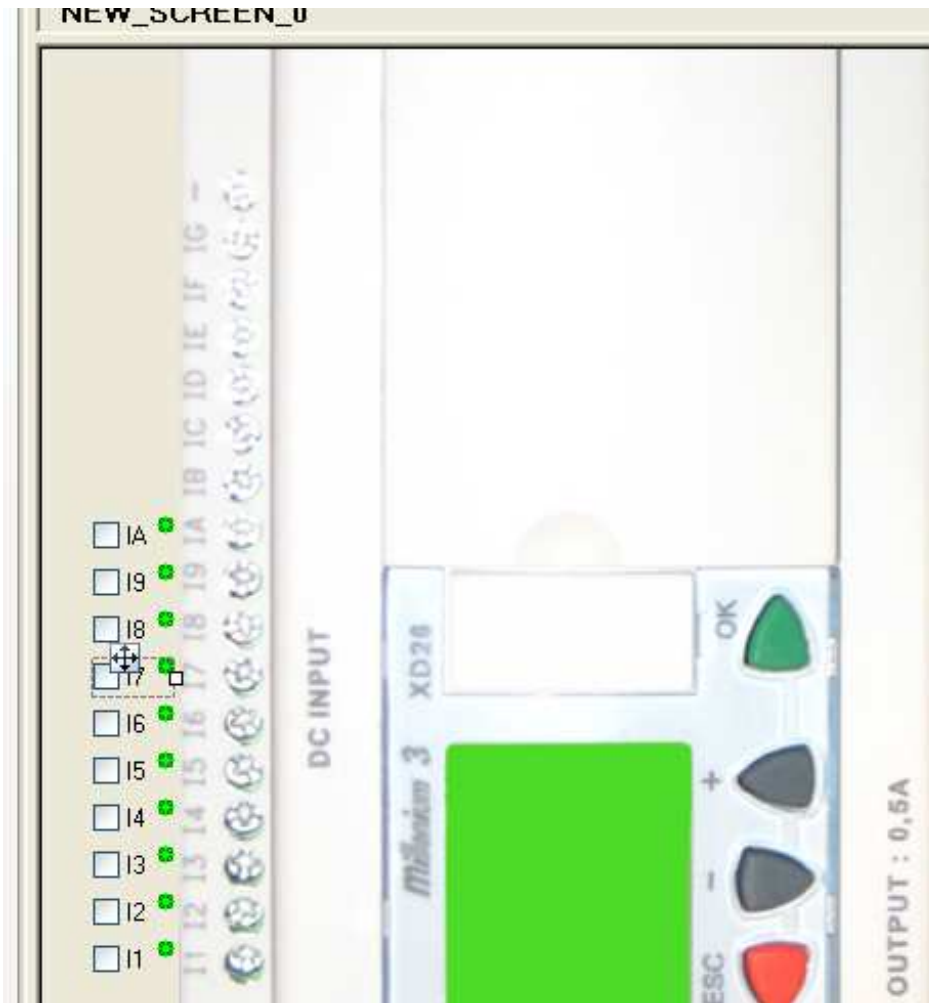
Read/Write = Read Multiple Register.

Starting Address = 20

Number of register = 1

TCPMB\_OUT\_REG\_20 register splitted into 16 one bit value. (as in wizard tutorial)

Now, needed frames are created, and we can add more values in screen, jump to screen window and add new check boxes for each controller discret inputs (I1 to IA). See following screen shot:



Here is association table for controller inputs.

TCP modbus Wizard:

- TCPMB\_OUT\_REG\_20\_B1: Controller I1
- TCPMB\_OUT\_REG\_20\_B2: Controller I2
- TCPMB\_OUT\_REG\_20\_B3: Controller I3
- TCPMB\_OUT\_REG\_20\_B4: Controller I4
- TCPMB\_OUT\_REG\_20\_B5: Controller I5
- TCPMB\_OUT\_REG\_20\_B6: Controller I6
- TCPMB\_OUT\_REG\_20\_B7: Controller I7
- TCPMB\_OUT\_REG\_20\_B8: Controller I8
- TCPMB\_OUT\_REG\_20\_B9: Controller I9
- TCPMB\_OUT\_REG\_20\_B10: Controller IA

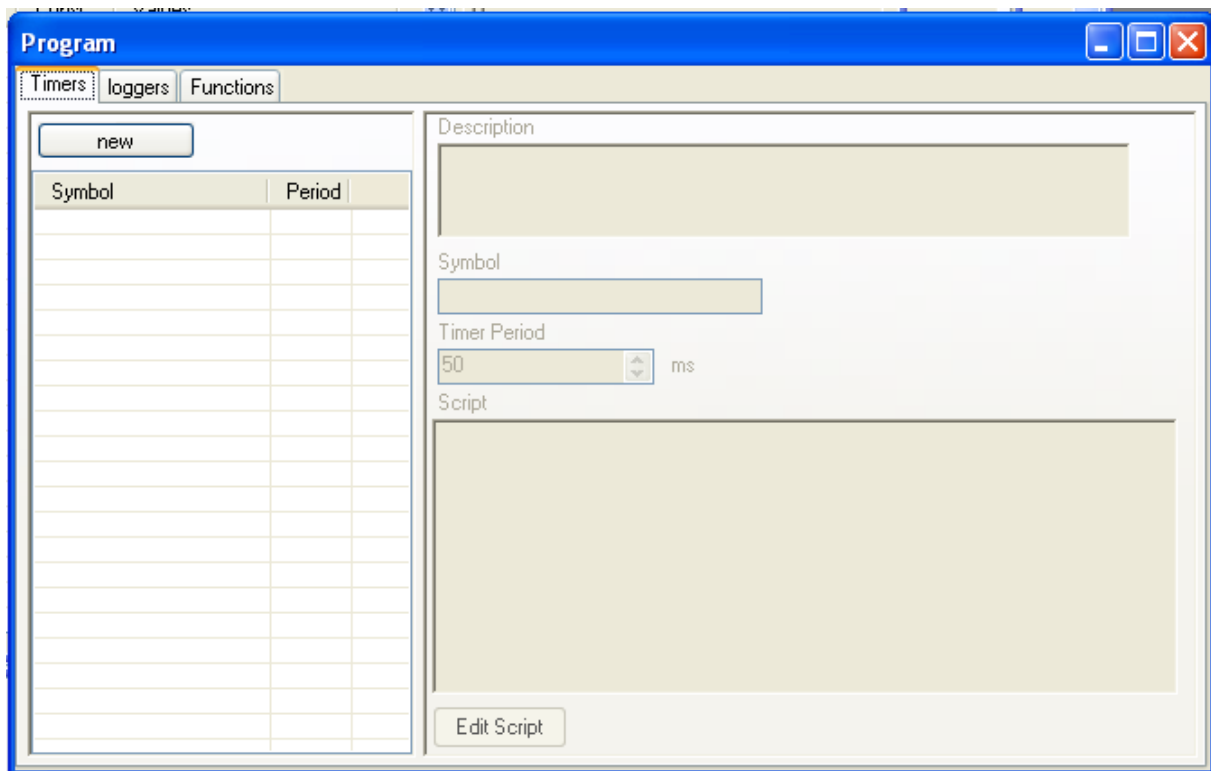
M3 SL Bloc Wizard:

- M3\_SLOUT\_I1\_B1: Controller I1
- M3\_SLOUT\_I1\_B2: Controller I2
- M3\_SLOUT\_I1\_B3: Controller I3
- M3\_SLOUT\_I1\_B4: Controller I4

M3\_SLOUT\_I1\_B5: Controller I5  
M3\_SLOUT\_I1\_B6: Controller I6  
M3\_SLOUT\_I1\_B7: Controller I7  
M3\_SLOUT\_I1\_B8: Controller I8  
M3\_SLOUT\_I1\_B9: Controller I9  
M3\_SLOUT\_I1\_B10: Controller IA

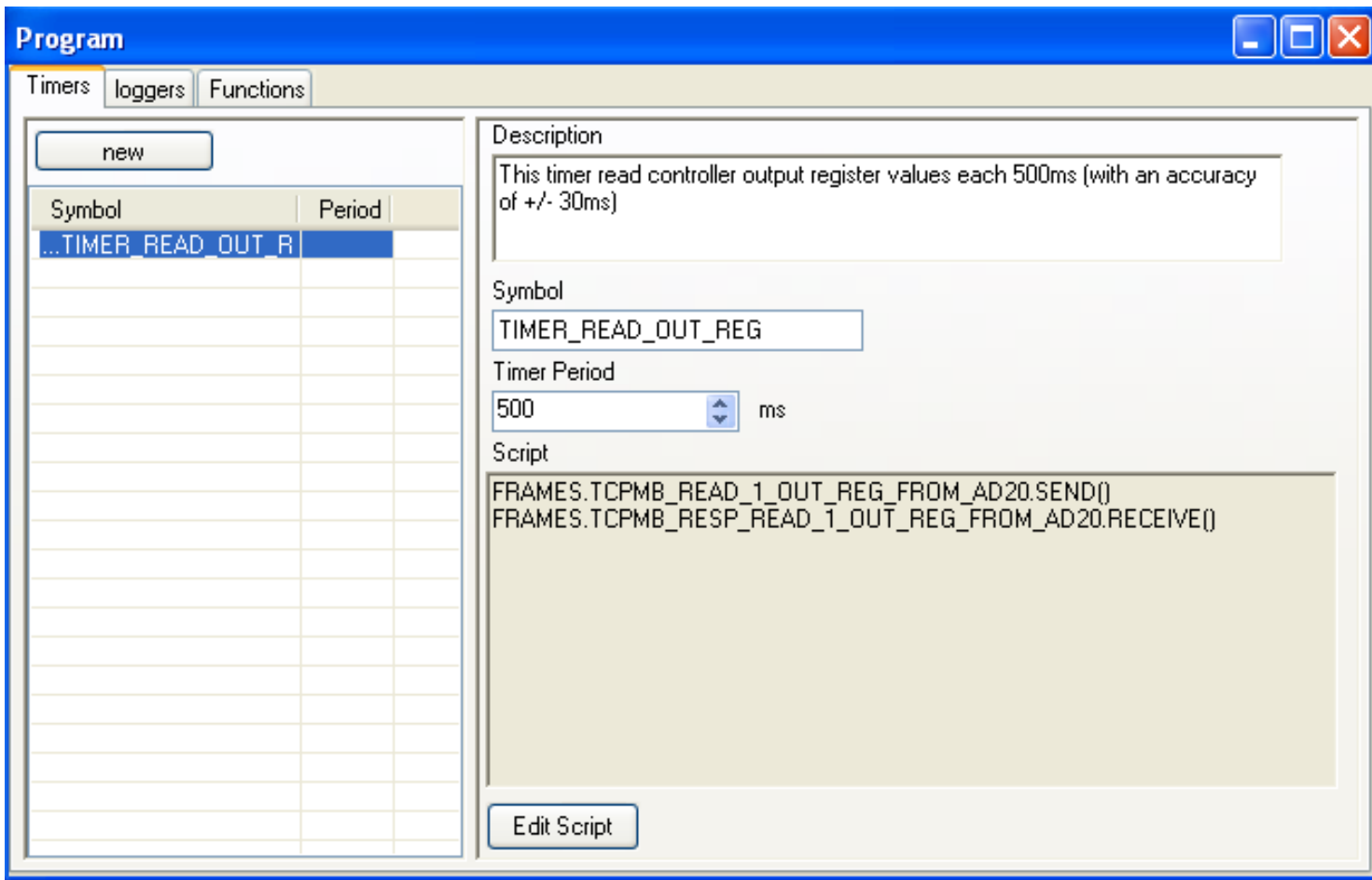
Once you've finished with screen configuration, active « Program Window ».

## 9. Program Window



This window is decomposed in 3 pages, one for each kind of item.  
The first page is the Timer Page:

### 9.1 Timers Page



There is no many thing to explain here. Timers objects execute their script periodically according to their period (in ms).

Timers are started or stopped when you press the « Start/Stop » button in Smart Command software. Then can also be started and stopped by script.

Script lines for starting or stopping timers are:

TIMERS.[TIMER SYMBOL].START()  
or TIMERS.[TIMER SYMBOL].STOP()

For our example, the script must read controller output register. So the script should be:

For M3 SL Bloc:

**FRAMES.M3\_READ\_SLOUT\_BLOC\_1.SEND()**  
**FRAMES.M3\_RESP\_READ\_SLOUT\_BLOC\_1.RECEIVE()**

For TCP Modbus:

**FRAMES.TCPMB\_READ\_1\_OUT\_REG\_FROM\_AD20.SEND()**  
**FRAMES.TCPMB\_RESP\_READ\_1\_OUT\_REG\_FROM\_AD20.RECEIVE()**

Do not forget to change timer period to 500ms (minimum timer period is 50 ms).

**9.2 Loggers Page**

**9.3 Functions Page**