

Smart Application tutoriel

Introduction:

Smart Application est un logiciel de supervision simple conçu pour les automates de type Smart Relays. Il a été testé avec les automates millennium 3 (Crouzet) en utilisant une liaison série standard, une liaison bluetooth et une liaison ethernet (avec l'extension XN05), et avec des modules d'entrée sortie déporté Advantys OTB en Ethernet .

INFORMATIONS IMPORTANTES:

Ce développement à été réalisé en dehors de toutes activités professionnelles sans le support de Crouzet ©. Crouzet © ou le développeur ne peuvent être tenu pour responsable pour tout dommage ou dysfonctionnement qui interviendrait lors de l'utilisation de ce logiciel développé dans le cadre d'activités privées.

Ce Document va vous expliquer comment utiliser ce logiciel afin de superviser un contrôleur Millennium 3 par liaison série (ou Bluetooth) et par liaison Ethernet.

Équipements requis:

- 1 contrôleur Millennium 3 (pour utiliser la liaison Ethernet il doit être extensible comme un XD10(S) 24 VDC or XD26(S) 24VDC)
- 1 extension Ethernet XN05 pour la liaison Ethernet.
- 1 ordinateur sous Windows avec le framework.Net Microsoft© (version 2,0 ou supérieur).
- 1 clé Bluetooth pour Millennium 3 (N° 88 970 104 0625) pour la liaison par Bluetooth.
- 1 câble série pour Millennium 3 pour la liaison série.
- Le logiciel CLS M3 2.1.0 ou version supérieur.

Installation:

Lancez l'exécutable Setup.exe et suivez les instructions. Une fois l'installation terminée, deux raccourcis sont ajoutés au bureau: un pour SmartConfig et un pour SmartCommand

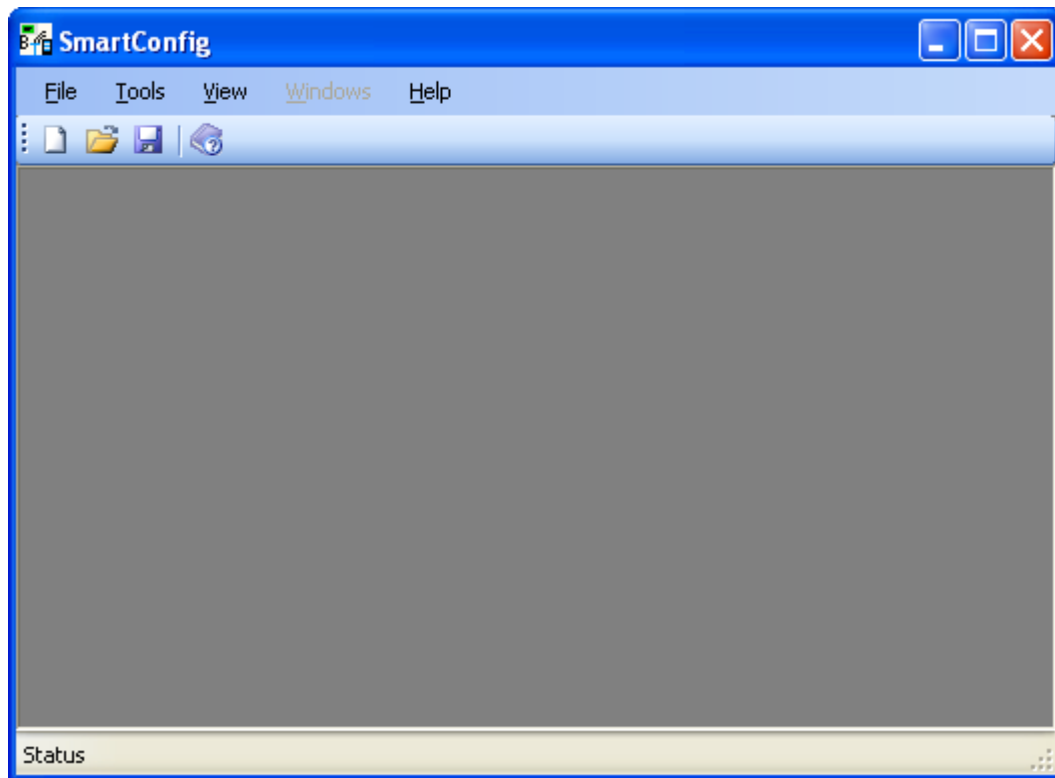
Informations:


Ce logiciel est conçu pour être utilisé par des utilisateurs ayant différents niveaux de connaissance en matière de communication avec des appareils distants. Les textes en **orange** peuvent être ignorés ou rester incompris par les simples utilisateurs des contrôleurs M3 sans que cela n'ait d'impacts sur leur utilisation de ce logiciel.

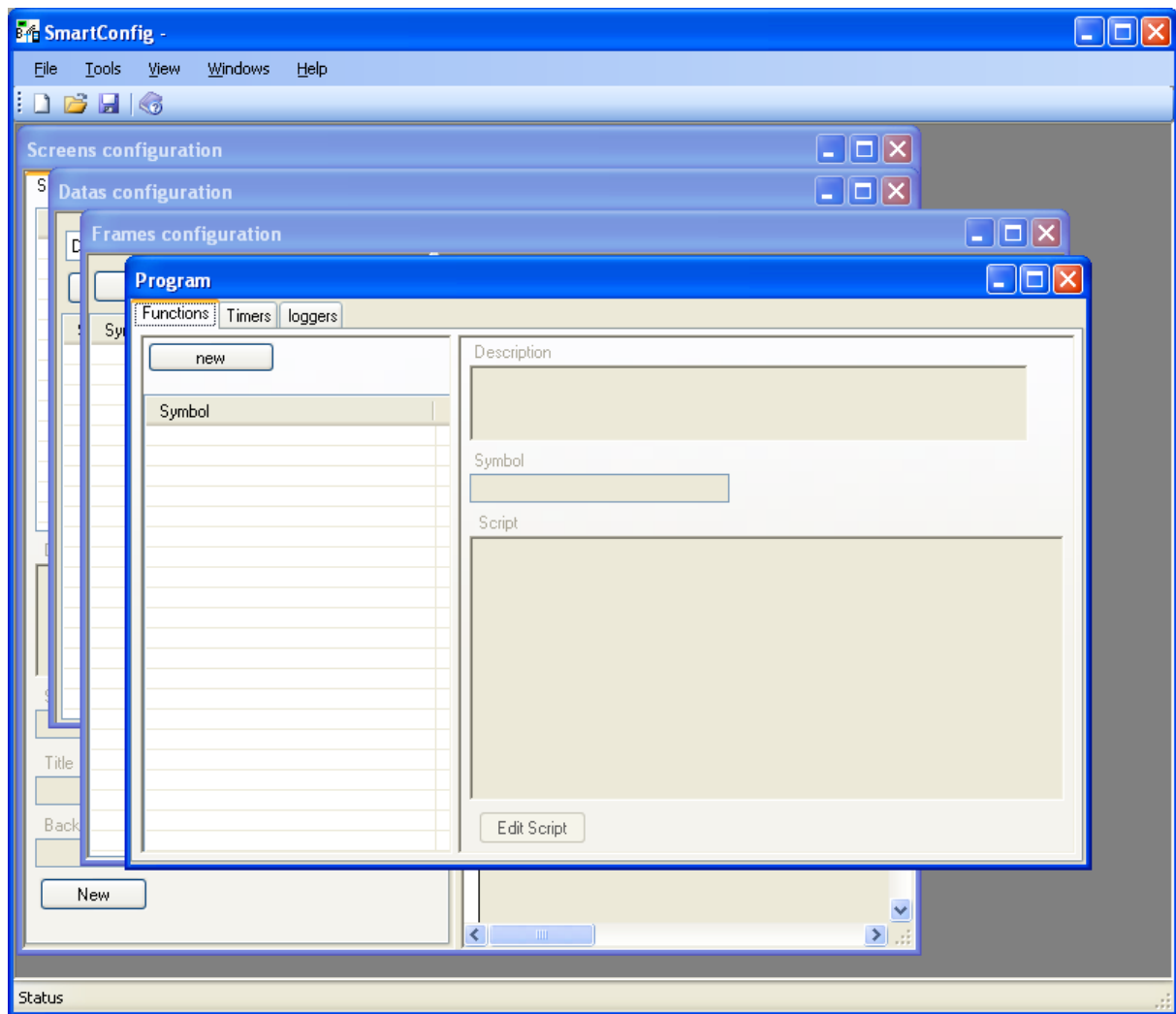
1. Création d'un fichier de supervision.....	3
2. Frames Wizards.....	4
2.1 Wizard SL Bloc M3.....	6
2.2 TCP Modbus Wizard.....	9
3. Frame Window (Fenêtre des trames).....	13
3.1 Frame properties(propriétés de la trame).....	14
3.2 Frame datas.....	16
4. Data Window (fenêtre des données).....	17
5. Screen Window (fenêtre des écrans).....	20
5.1 Screens Page.....	21
5.2 Designer.....	22
5.2 “Tool” page (page des outils).	22
5.3 Control Options Page.....	26
Associer les données et les contrôles.....	27
5.4 Item Script Page (Page du script des objets).....	30
6. Script Editor (l'éditeur de script).....	31
7. SmartCommand.....	36
7.1 Connection configuration (configuration de la connexion).....	36
7.2 Démarrer votre supervision.....	38
8.Finaliser la Supervision de votre M3	39
9. Program Window.....	41
9.1 Timers Page (page des Timers).....	42
9.2 Loggers Page (page des Loggers).....	43
9.3 Functions Page (page des fonctions).....	45
10 ANNEXES : Les contrôles avancés.....	46
10.1 le rectangle coloré.....	46
10.2 l'ellipse coloré.....	46
10.3 la jauge.....	46
10.4 le serpent.....	46
10.5 le contrôle 2 images.....	46
10.6 le contrôle 4 images.....	46
10.7 l'afficheur formaté.....	46
10.8 le ballon d'eau.....	46

1. Création d'un fichier de supervision

Lancez SmartConfig, la fenêtre suivante apparaît.



Cliquez sur le bouton « new »  ou sur « new » dans le menu « File ». 4 nouvelles fenêtre apparaissent.



Avant de décrire l'utilisation de chaque fenêtre nous allons voir comment utiliser le générateur de trame automatique.

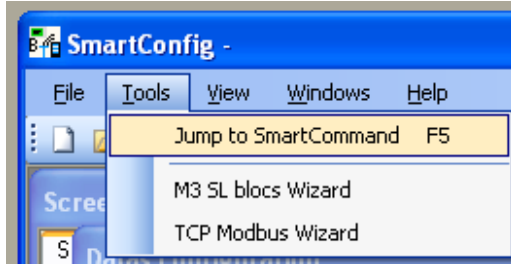
2. Frames Wizards

GENERALITE SUR LES TRAMES (« Frame » en anglais):

Il est très important de comprendre la notion de trame, sans en connaître les détails. Lorsqu'on souhaite communiquer entre deux machines, il est nécessaire d'échanger des messages (les trames) qui fonctionnent sous forme de requête / réponse, dans le cas du dialogue avec les automates, comme lorsque vous surfez sur internet. Par exemple lorsque vous allez sur un site et cliquez sur un lien, votre ordinateur envoie un message au serveur « peut tu m'envoyer cette page » et ensuite le serveur répond « tiens voilà la page », votre ordinateur l'affiche ensuite dans votre navigateur. Dans le cas du dialogue avec un automate distant, le principe est le même, sauf que les questions seront du type « peut tu me donner cette (ces) valeur(s) » et l'automate répond « tiens voilà la (les) valeur(s) que tu m'as demandé ». Bien entendu en cas d'erreur, il peut également répondre « nan je ne peux pas te la donner » dans le cas où la donnée n'existerait pas ou si elle n'est pas disponible.

Ceci montre donc qu'un échange complet est toujours au moins composé de deux trames : la requête, et la réponse. Les deux trames nécessaires à chaque échange seront automatiquement générées par les wizards.

Ouvrez le menu "Tool".



Dans ce menu vous pouvez trouver 3 commandes:

- **Jump to SmartCommand:** Ce menu permet de lancer immédiatement l'application SmartCommand avec le fichier courant nous verrons l'utilisation de SmartCommand beaucoup loin dans le tutoriel.
- **M3 SL bloc Wizard:** Ce menu vous permet d'afficher l'assistant de génération de trames d'échange pour la liaison série (ou Bluetooth) avec les SL BLOC.
- **TCP Modbus Wizard:** Ce menu vous permet d'afficher l'assistant de génération de trames d'échange TCP/Modbus pour la supervision via l'extension XN05 (Ethernet).

2.1 Wizard SL Bloc M3.

Si vous souhaitez utiliser ce logiciel en utilisant l'extension XN05 , allez directement ici [Wizard TCP Modbus.](#)

Ouvrez le « M3 SL Bloc Wizard ». La fenêtre suivante apparaît.

WizardSLForm

Frame type
M3 SL Input Bloc

Read / Write
Read

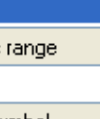
Address range
1-8

Frame symbol
M3_READ_SLIN_BLOC_1

☒ Generate answer frame

Answer frame symbol
M3_RESP_READ_SLIN_BLOC_1

M3 SL Bloc Bitmap



Data symbol	Data size	SL Input/Duput 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

OK Cancel Split selected data Join selected datas

La partie gauche de la fenêtre permet de configurer la trame:

- **Frame Type:** Choisissez si vous voulez utiliser une trame de lecture ou d'écriture. L'image correspondante au bloc SL de SLCM3 apparaît en bas de la fenêtre. Il est mis à jour en fonction des différents paramètres.
- **Read/Write:** Choisissez si vous souhaitez lire ou écrire dans le bloc SL.
- **Address Range:** Sélectionnez la plage d'adresse du bloc SL en fonction de la configuration de votre programme M3.
- **Frame Symbol:** Symbole (chaîne de caractère) de la trame qui va être créé pour envoyer la requête au contrôleur.
- **Answer frame Symbol:** Symbole de la trame de réponse qui sera envoyé par le contrôleur en réponse à la requête.

Laissez les paramètres tel que et continuez.

La partie droite de la fenêtre est la liste des données qui vont être utilisée pour stocker/échanger les valeurs contenues dans le contrôleur.

Cette liste vous permet de découper chaque donnée en plusieurs données de tailles plus petites jusqu'à avoir 16 données de 1 bit pour un mot de 16 bits (en passant par tout les intermédiaires, 8bits, 4bits et 2 bits).

Voyons concrètement comment cela est utilisé avec le fichier **Serial Example (SL Bloc).pm3**

The diagram illustrates a 16-bit parallel adder implemented using two 8-bit adders. On the left, an 8-bit adder is labeled 'SL' with a blue '1 → 8' indicating its 8-bit width. It has eight green input pins and one green output pin. On the right, another 8-bit adder is labeled 'DEC' with a blue '16 → 8' indicating its 8-bit width. It has eight green input pins and one green output pin. The output of the 'SL' adder is connected to the input of the 'DEC' adder. The 'DEC' adder has a 'BIN' input and a 'DEC' output. The 'DEC' output is connected to the input of the 'SL' adder. The 'BIN' input of the 'DEC' adder is connected to the output of the 'SL' adder. The 'DEC' adder has eight green output pins, each connected to a blue output pin labeled 'DO'. The 'SL' adder has one green output pin connected to a blue output pin labeled 'DO'. The outputs are labeled O1 through O8, each with a blue 'DO' label and a blue waveform icon.

[illegible]

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_O1_MSB	8	1
M3_SLIN1_O1_LSB	8	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

Chaque donnée crée prend un nom spécifique en fonction de sa taille et de son entrée/sortie correspondante sur le bloc SL

Ici, les deux données nommées M3_SLIN1_O1_MSB (MSB pour Most Significant Byte) et M3_SLIN1_O1_LSB (LSB pour Less Significant Byte).

Sélectionnez maintenant la donnée M3_SLIN1_O1_LSB et cliquez sur le bouton “Split” (ou pressez sur la touche “s” du clavier).

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_O1_MSB	8	1
M3_SLIN1_O1_B5-8	4	1
M3_SLIN1_O1_B1-4	4	1

Cette fois, la donnée de 8 bits a été découpée en deux données de 4 bits.

Une fois de plus les données créées on pris le nom correspondant à leur taille et à leur sortie

- M3_SLIN1_O1_B5-8 pour la donnée commençant au 5ième bit et finissant au 8ième bit de la sortie du bloc SL.
- M3_SLIN1_O1_B1-4 for data pour la donnée commençant au 1er bit et finissant au 4ième bit de la sortie du bloc SL.

Continuez de découper chaque données commençant par “M3_SLIN1_O1” jusqu'à obtenir 16 données de 1 bit (une pour chaque bit de la sortie de bloc).

Data symbol	Data size	SL Input/Output n°
M3_SLIN1_O1_B16	1	1
M3_SLIN1_O1_B15	1	1
M3_SLIN1_O1_B14	1	1
M3_SLIN1_O1_B13	1	1
M3_SLIN1_O1_B12	1	1
M3_SLIN1_O1_B11	1	1
M3_SLIN1_O1_B10	1	1
M3_SLIN1_O1_B9	1	1
M3_SLIN1_O1_B8	1	1
M3_SLIN1_O1_B7	1	1
M3_SLIN1_O1_B6	1	1
M3_SLIN1_O1_B5	1	1
M3_SLIN1_O1_B4	1	1
M3_SLIN1_O1_B3	1	1
M3_SLIN1_O1_B2	1	1
M3_SLIN1_O1_B1	1	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

Ouvrez une fois de plus le **M3 SL Bloc Wizard** et modifiez uniquement le champ “Read/Write” en sélectionnant le choix “Write”. Découpez la première donnée de sortie de 16 bits en 16 données de 1 bit comme nous l'avons fait précédemment puis cliquez sur le bouton « OK ». puis allez directement à la [Fenêtre de description des trames](#).

- **Register Type:** Sélectionnez le type de registre que vous souhaitez lire/écrire. Cette propriété est principalement utilisés pour créer les noms de donnés en fonction du type de registre. Notez que les registres de sortie ne sont accessibles qu'en lecture.
- **Read/Write:** Choisissez si vous souhaitez lire ou écrire dans le bloc SL.
- **Starting register address:** Sélectionnez l'adresse de départ du(des) registre(s) qui vont être lut ou écrit.
- **Number of registers:** Disponible uniquement pour les commandes modbus “Read holding registers” et “Write multiple registers”. Permet de choisir le nombre de registres qui vont être lut ou écrit.

- **Frame Symbol:** Symbole (chaîne de caractère) de la trame qui va être créé pour envoyer la requête au contrôleur.
- **Answer frame Symbol:** Symbole de la trame de réponse qui sera envoyé par le contrôleur en réponse à la requête.

Laissez les paramètres tel que et continuez.

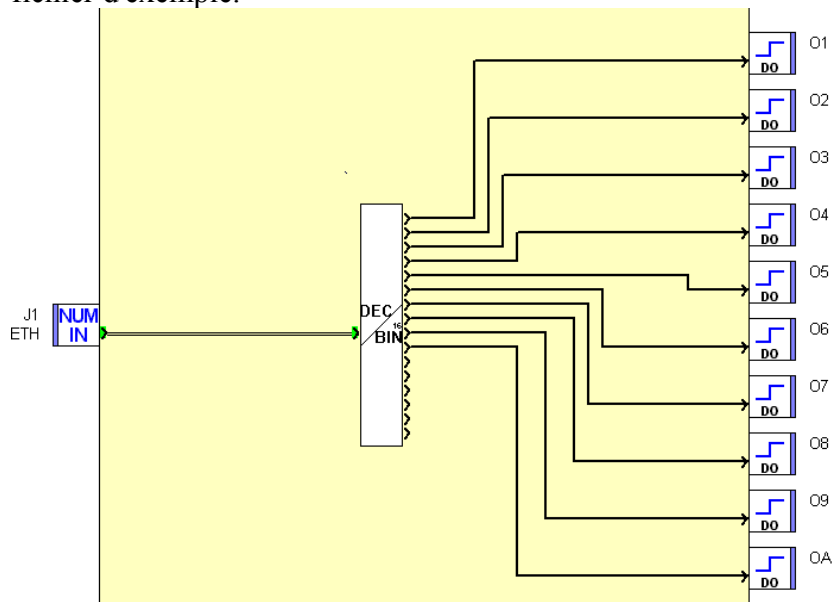
NOTE: l'extension M3 XN05 supporte uniquement les commandes “Write simple register” et “Read Holding register”.

La partie droite de la fenêtre est la liste des données qui vont être utilisée pour stocker/échanger les valeurs contenues dans le contrôleur.

Cette liste vous permet de découper chaque donnée en plusieurs données de tailles plus petites jusqu'à avoir 16 données de 1 bit pour un mot de 16 bits (en passant par tout les intermédiaires, 8bits, 4bits et 2 bits).

Voyons concrètement comment cela est utilisé avec le fichier **Ethernet Example (XN05).pm3**

Voici une capture d'écran du câblage du bloc d'entrée NUM IN (plot d'entrée J1) dans le fichier d'exemple.



La sortie du bloc est connecté à un convertisseur DEC/BIN qui permet d'utiliser chaque bit de la sortie 1 du bloc SL pour contrôler les sorties relais du contrôleur.

Nous souhaitons donc utiliser chaque bit de la sortie comme étant une donnée indépendante.

Configurez les paramètres du wizard comme sur l'image suivante

Cliquez sur la donnée nommée TCPMB_IN_REG_12 et cliquez sur le bouton « Split selected data »

La donnée de 16 bits est découpé en deux autres données de 8 bits. Qui sont définie sur le même registre (même sortie de bloc).

Chaque donnée créée prend un nom spécifique en fonction de sa taille et de son registre. Ici, les deux données sont nommées TCPMB_IN_REG_12_MSB (MSB pour Most Significant Byte) et TCPMB_IN_REG_12_LSB (LSB pour Less Significant Byte). Maintenant sélectionnez la donnée TCPMB_IN_REG_12_LSB et cliquez sur le bouton “Split” (ou pressez la touche “s” du clavier).

Data symbol	Data size	Register number
TCPMB_IN_REG_12_MSB	8	1
TCPMB_IN_REG_12_B5-8	4	1
TCPMB_IN_REG_12_B1-4	4	1

Cette fois, la donnée de 8 bits a été découpée en deux données de 4 bits.

Une fois de plus les données créées on a pris le nom correspondant à leur taille et à leur registre

- TCPMB_IN_REG_12_B5-8 pour la donnée commençant au 5^{ème} bit et finissant au 8^{ème} bit de la sortie du bloc.
- M3_SLIN1_O1_B1-4 for data pour la donnée commençant au 1^{er} bit et finissant au 4^{ème} bit de la sortie du bloc.

Continuez de découper chaque données commençant par “TCPMB_IN_REG_12” jusqu'à obtenir 16 données de 1 bit (une pour chaque bit de la sortie du bloc) comme décrit dans la capture d'écran suivante.

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

Nous avons maintenant 16 données de 1 bit pour chaque bit du registre. La trame est maintenant prête à être créée. Cliquez sur le bouton « OK ».

Avant de voir ce qui a été généré, nous allons créer la trame d'écriture du bloc SL 1-8.

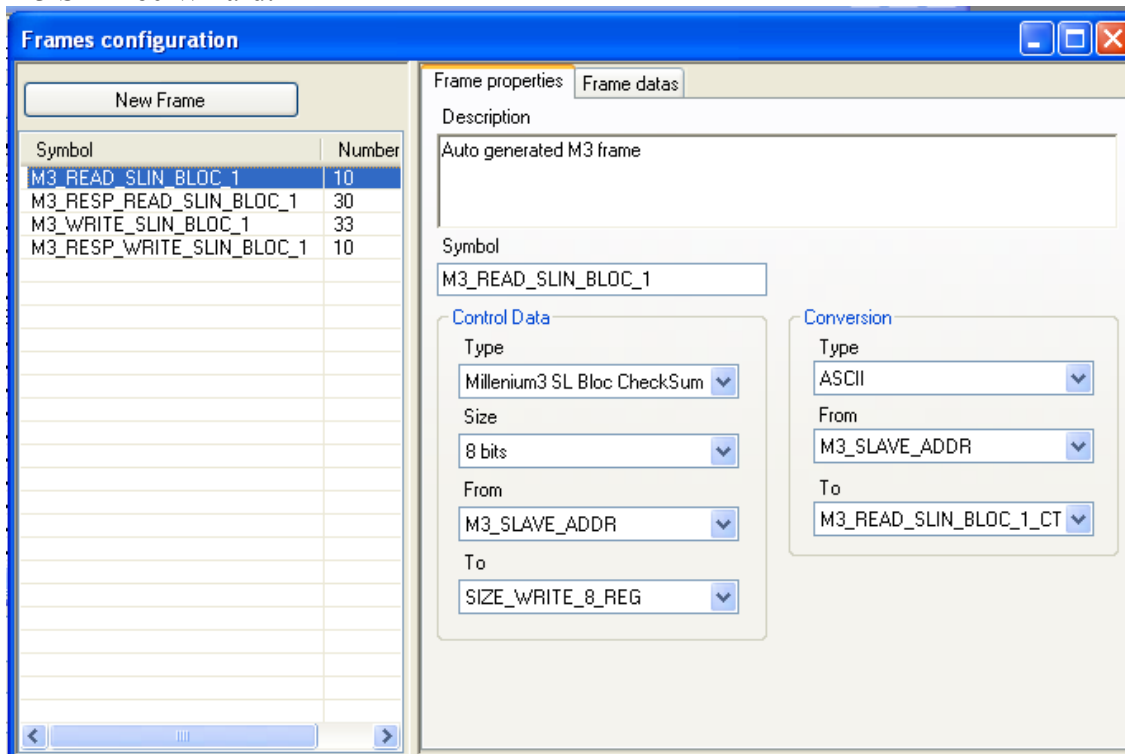
Ouvrez une fois de plus le **TCP Modbus Wizard** et sélectionnez les mêmes paramètres que précédemment en modifiant uniquement le champ “Read/Write” en sélectionnant le choix “Write simple register”. Découpez la première donnée de sortie de 16 bits en 16 données de 1 bit comme nous l'avons fait précédemment puis cliquez sur le bouton « OK ».

Continuez le tutoriel afin de visualiser ce qui a été créé.

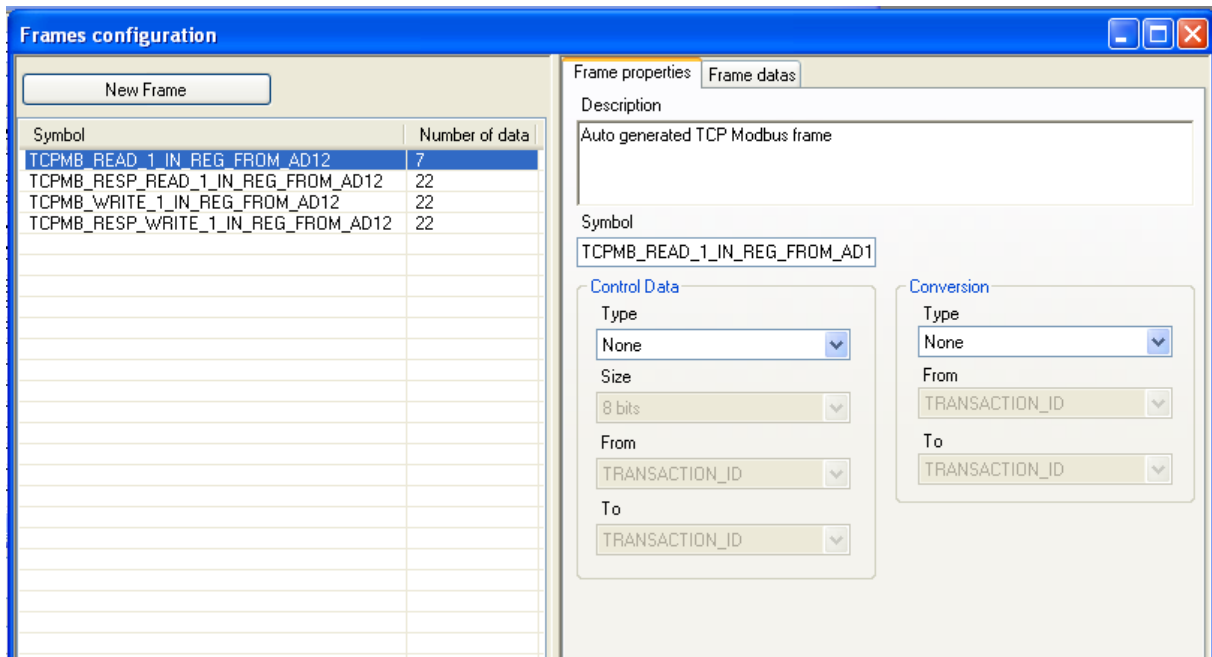
3. Frame Window (Fenêtre des trames)

Activez la fenêtre des trames « Frame Configuration » dans le logiciel SmartConfig.
En fonction du Wizard que vous avez utilisé, vous obtenez l'un des résultats suivants:

M3 SL Bloc Wizard:



TCP Modbus Wizard:



Dans la partie gauche de la fenêtre, vous avez la liste des trames existantes, celles que vous venez juste de générer automatiquement sont ici.

Dans la partie droite de la fenêtre, vous avez les propriétés des trames ainsi que les données des trames (respectivement « Frame Properties » et « Frame Datas »)

3.1 Frame properties(propriétés de la trame)

Frame properties | Frame datas

Description
Auto generated M3 frame

Symbol
M3_WRITE_SLIN_BLOC_1

Control Data

Type
Millenium3 SL Bloc CheckSum

Size
8 bits

From
M3_SLAVE_ADDR

To
M3_SLIN1_08

Conversion

Type
ASCII

From
M3_SLAVE_ADDR

To
M3_WRITE_SLIN_BLOC_1_C

Sur cette page vous pouvez trouver les paramètres suivants:

- **Description:** description de la trame
- **Symbol:** symbole de la trame

NOTE: la partie qui suit n'est utile et compréhensible que par les personnes maîtrisant les bases des protocoles de communication. Si vous êtes un simple utilisateur du M3, vous pouvez faire l'impasse sur cette partie et sauter directement au chapitre 4, car il vous est seulement nécessaire de connaître l'existence des trames.

- Control Data group:

Comme beaucoup de protocoles de communication ont besoin d'une donnée de contrôle, vous pouvez ici définir le type de donnée de contrôle qui sera utilisé par la trame. Lorsque vous sélectionnez une When donnée de control d'un type différent de « none », une donnée spéciale va être créée automatiquement et ajoutée à la trame, vous pouvez ensuite déplacer cette donnée à la bonne position dans la trame pour respecter le protocole voulu. Pour l'instant, seuls les données de contrôles pour le protocole spécifique de la liaison série du M3 ainsi que le CRC16 du modbus.

NOTE: Si vous voulez utiliser ce logiciel avec un contrôleur / appareil distant qui demande un checksum ou CRC spécifique, vous pouvez me demander de l'ajouter, à condition que vous me donniez son algorithme de calcul ainsi que ses règles.

- **Type:** type de la donnée de contrôle, les choix disponibles sont « none », « M3SLBloc checksum » et modbus CRC16.
- **Size:** taille en bits de la donnée de contrôle.

- **From/To:** Certains protocoles calculent la valeur de la donnée de contrôle sur uniquement une partie de la trame, donc ici vous pouvez définir la première et la dernière donnée prise en compte dans le calcul du checksum/CRC. Si la donnée sélectionnée est d'une taille inférieure à un octet, le calcul du checksum ou CRC commence dans l'octet où se trouve la donnée « From Data » et se termine à l'octet où se trouve la donnée « To data ».

- **Conversion Group:**

Certains protocoles de communication utilisent une conversion de la trame brute. Comme pour la donnée de contrôle, vous pouvez définir ici le type de conversion que vous souhaitez utiliser, ainsi que le début et la fin de la conversion.

- **Type:** Pour le moment, le seul mode de conversion disponible est la conversion « ASCII » (utilisé pour la communication série du M3).

- **From/To:** fonctionne comme pour la donnée de contrôle.

3.2 Frame datas

Frame properties			
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_Q_ADDR	8	True	0
SIZE_WRITE_Q_REG	8	True	16
M3_SLIN1_O1_B16	1	False	0
M3_SLIN1_O1_B15	1	False	0
M3_SLIN1_O1_B14	1	False	0
M3_SLIN1_O1_B13	1	False	0
M3_SLIN1_O1_B12	1	False	0
M3_SLIN1_O1_B11	1	False	0
M3_SLIN1_O1_B10	1	False	0
M3_SLIN1_O1_B9	1	False	0
M3_SLIN1_O1_B8	1	False	0
M3_SLIN1_O1_B7	1	False	0
M3_SLIN1_O1_B6	1	False	0
M3_SLIN1_O1_B5	1	False	0
M3_SLIN1_O1_B4	1	False	0
M3_SLIN1_O1_B3	1	False	0
M3_SLIN1_O1_B2	1	False	0
M3_SLIN1_O1_B1	1	False	0
M3_SLIN1_O2	16	False	0
M3_SLIN1_O3	16	False	0
M3_SLIN1_O4	16	False	0
M3_SLIN1_O5	16	False	0
M3_SLIN1_O6	16	False	0
M3_SLIN1_O7	16	False	0
M3_SLIN1_O8	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

Sur cette page, vous pouvez trouver la liste des données de la trame sélectionnée. Ici vous pouvez modifier l'ordre des données par glisser/déposer dans la liste.

ATTENTION: Vous ne devez pas modifier les trames qui ont été générés automatiquement car elle ont été définies et testés spécialement pour les protocoles de la liaison série M3 et TCP modbus. Cependant, si vous maîtrisez les protocoles de communications et leur règles, vous pouvez modifier l'ordre des données ou les propriétés d'une trame.

4. Data Window (fenêtre des données)

Activez la fenêtre de configuration des données (« Data Configuration ») dans l'application SmartConfig.

Lorsque vous avez utilisé les Wizard pour créer des trames, toutes les données nécessaires ont été automatiquement générées et ajoutées ici. En fonction du type de Wizard que vous avez utilisé, vous obtenez l'un des résultats suivants.

M3 SL Bloc Wizard:

Datas configuration

Default group

New Data Manage Groups

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
M3_END_OF_FRAME2	8	True	Def: 10, Min: 0, Max: 255
M3_SLIN1_O1_B16	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B15	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B14	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B13	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B12	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B11	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B10	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B9	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B8	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B7	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B6	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B5	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B4	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B3	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B2	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O1_B1	1	False	Def: 0, Min: 0, Max: 1
M3_SLIN1_O2	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_O3	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_O4	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_O5	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_O6	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_O7	16	False	Def: 0, Min: -32768, Max: 32...
M3_SLIN1_O8	16	False	Def: 0, Min: -32768, Max: 32...
M3_MODBUS_WRITE_MR	8	True	Def: 16, Min: 0, Max: 255

Description

Symbol: M3_END_OF_FRAME2

Size: 8 bits data

Min: 0

Max: 255

Default: 10

☒ Constant

[illegible]

La liste affiche toutes les information sur chaque données. Les différents paramètres vont être détaillés dans la partie droite de la fenêtre.

The image shows a 'Description' dialog box with the following fields:

- Description:** A large empty text area.
- Symbol:** A text box containing 'TCPMB_IN_REG_12_B16'.
- Size:** A dropdown menu showing '1 bit data'.
- Min:** A spin box showing '0'.
- Max:** A spin box showing '1'.
- Default:** A spin box showing '0'.
- Constant:** An unchecked checkbox.

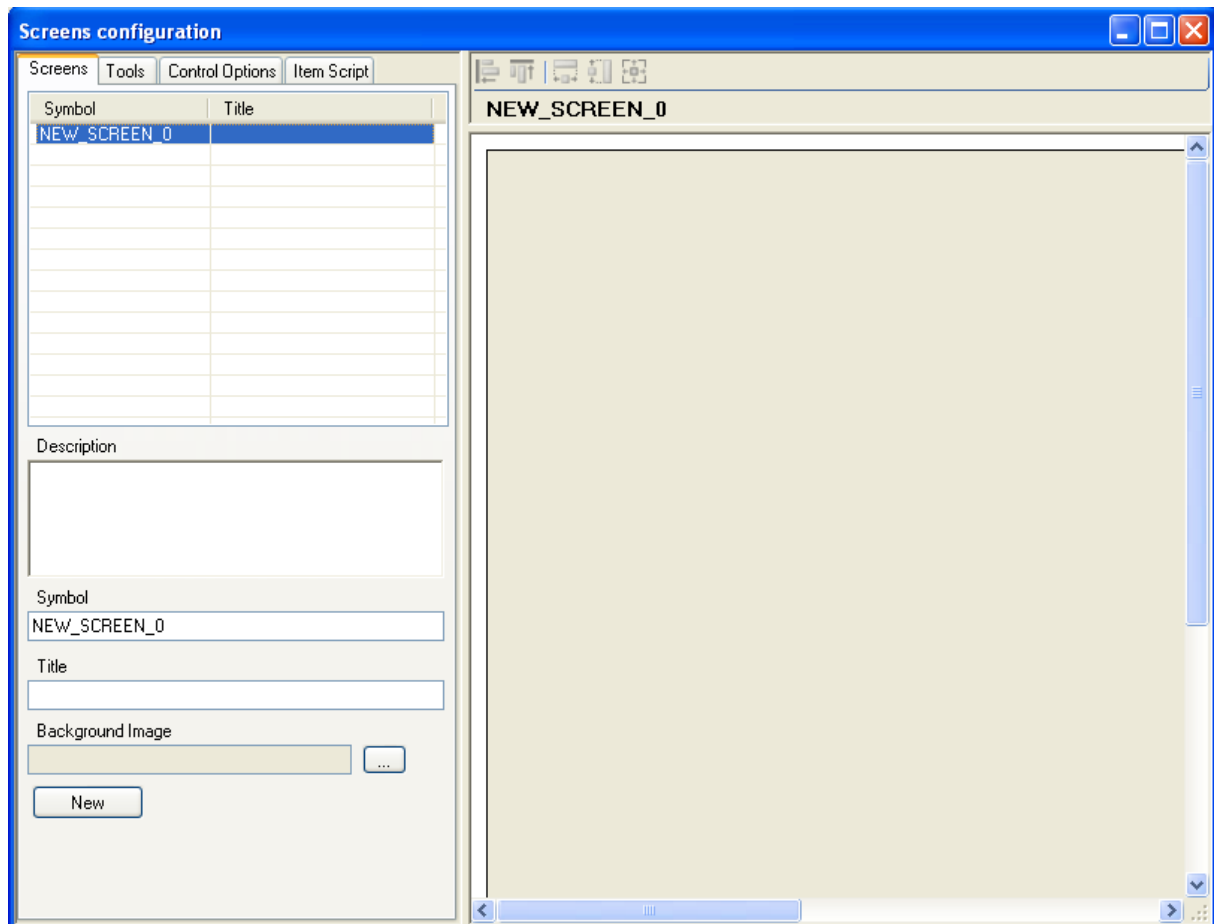
Dans la partie droite de la fenêtre vous pouvez trouver les paramètres suivants:

- **Description:** description de la donnée
- **Symbol:** symbole de la donnée
- **Size:** Taille en bits de la donnée
- **Max:** valeur maximum que la donnée peut prendre. La fourchette de valeur change en fonction de la taille de la donnée.
- **Min:** valeur minimum que la donnée peut prendre. La fourchette de valeur change en fonction de la taille de la donnée.
- **Default:** Valeur par défaut qu'aura la donnée à l'initialisation du logiciel en mode « Commande » (voir SmartCommand)
- **Constant:** Si cette option est cochée, la donnée sera utilisée en tant que constante et aura toujours sa valeur par défaut, si quelque chose tente de modifier sa valeur en mode « Commande », un message vous le notifiera dans la fenêtre de logs de l'application SmartCommand.

NOTE: Le paramètre “Constant” peut être critique car il est utilisé pour la réception d'une trame. Chaque donnée constante est testée à celle reçue et doit être égale, alors que les données non constantes sont assignés à la valeur reçue.

Le nommage par défaut des données indique la position de la donnée dans son registre/bloc SL. Ceci vous permet de repérer quelle donnée correspond à quoi, et donc d'y ajouter une description ou de modifier son symbole afin d'en faciliter l'utilisation par la suite.

5. Screen Window (fenêtre des écrans)



Dans cette fenêtre; vous allez créer et designer l'apparence et le comportement de vos propres fenêtres qui seront affichées dans l'application SmartCommand.

Ici, rien ne se fait tout seul car chaque écran est spécifique à l'utilisation qui lui est destinée.

5.1 Screens Page

Symbol	Title
NEW_SCREEN_0	

Description

Symbol
NEW_SCREEN_0

Title

Background Image

New

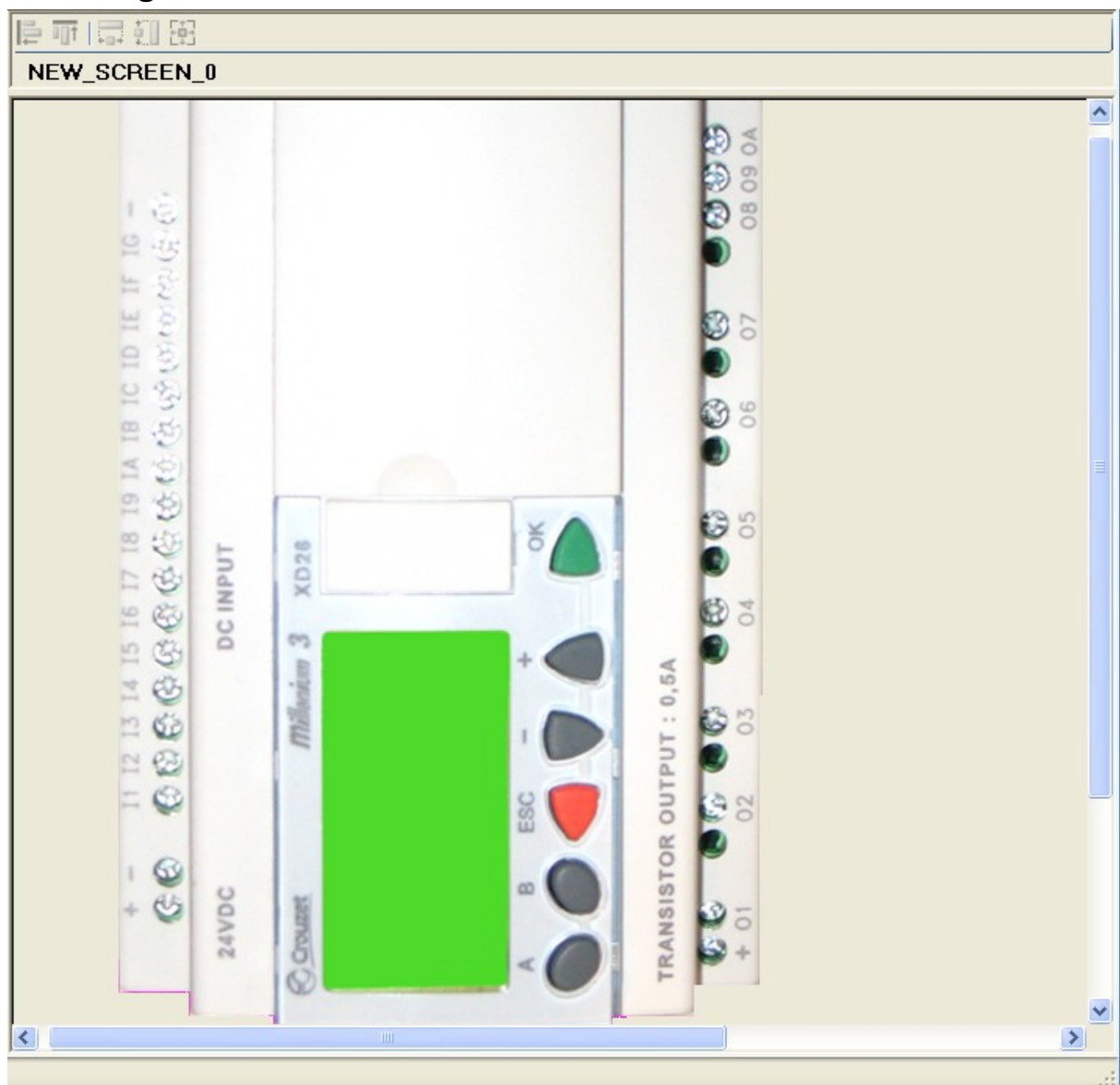
Dans la partie haute de cette page, vous pouvez trouver la liste des écrans existants. Cliquez sur le bouton « new » pour en créer un puis sélectionnez le.

Dans la partie basse vous pouvez trouver les propriétés suivantes.

- **Description:** description de l'écran
- **Symbol:** symbole de l'écran
- **Title:** Titre de l'écran, ce sera le titre de la fenêtre dans l'application SmartCommand.
- **Background Image:** Chemin du fichier qui sera utilisé en tant qu'image de fond pour l'écran sélectionné.

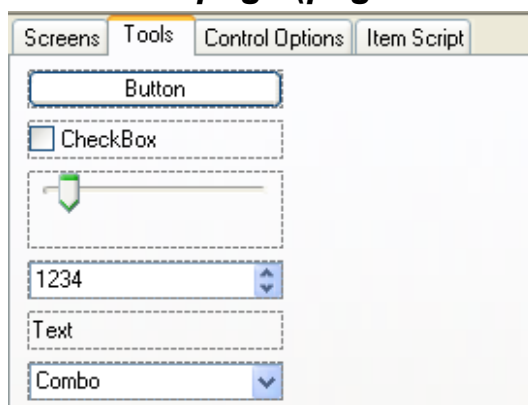
Entrez le titre « Millenium 3 Controller » et pressez le bouton « ... » afin de choisir l'image de fond. Une fenêtre de choix de fichier s'ouvre dans le répertoire d'installation de l'application. Allez dans le dossier « Examples » et choisissez l'image « **M3XD26S.bmp** ». Cliquez ensuite dans la partie droite de la fenêtre afin de valider les changements. L'image sélectionnée s'affiche en fond du designer d'interfaces.

5.2 Designer



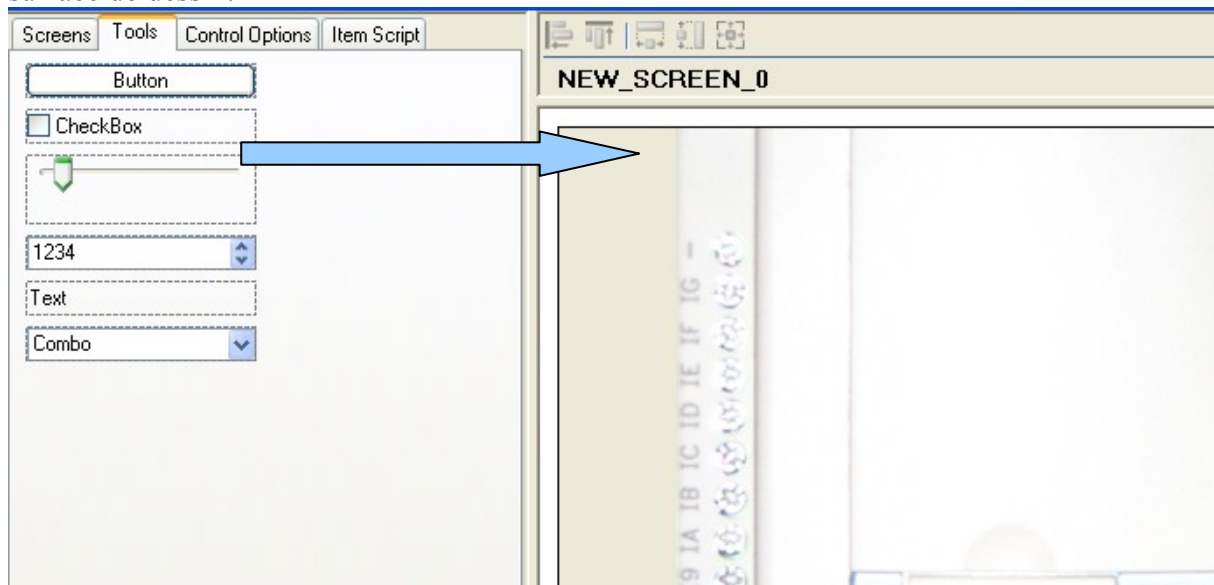
Voici le designer avec l'image fraîchement sélectionnée.

5.2 "Tool" page (page des outils).



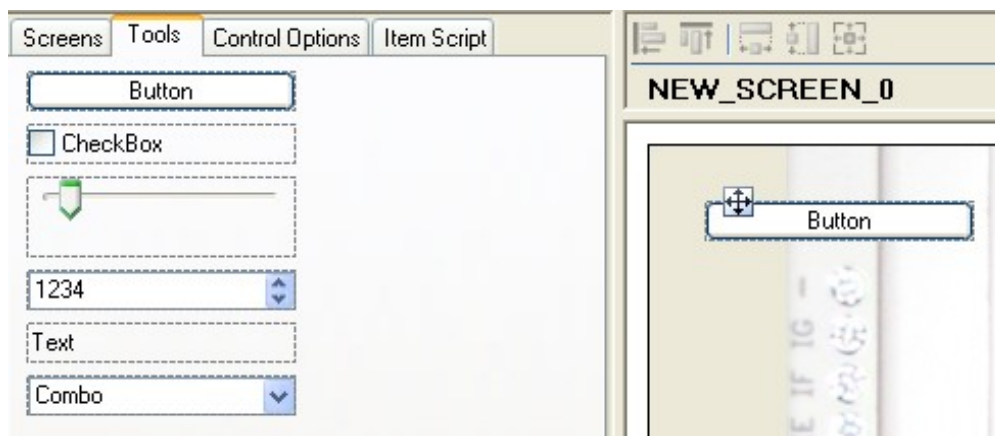
Dans cette page, vous trouverez les différents types de contrôles disponible qui peuvent être utilisés pour créer votre interface.

Pour les ajouter, il suffit de faire un « glisser / déposer » à partir de la page d'outils vers la surface de dessin.



Une fois qu'un contrôle a été ajouté, vous pouvez le déplacer et le redimensionner en le sélectionnant puis en utilisant le bouton de déplacement et les carrés de redimensionnement.

NOTE: certains types de contrôles ne peuvent être redimensionnés qu'en largeur.



Dans la capture d'écran nous ne pouvons voir que les contrôles standards. La liste des contrôles évolués est décrite en annexe de ce tutoriel. (fin du document)

Tout d'abord il est important de comprendre la notion de **donnée associée**:

une « donnée associée » est une donnée (décrite précédemment) qu'on associe à un contrôle.

Chaque contrôle possède un comportement spécifique avec la valeur de la donnée qui lui est associée. Par exemple, un bouton change la valeur de sa donnée associée lorsqu'on clique dessus en mode « Commande ». Par exemple, si la valeur de la donnée était 0 et qu'on clique sur le bouton, la valeur de la donnée passera à 1; si elle était à 1, la valeur passera à 0. Ceci permet typiquement d'envoyer une commande de type impulsion à l'automate, en connectant le bit correspondant à la donnée à un bloc timer SW (timer impulsif) dans le programme M3.

Description des contrôles standards:

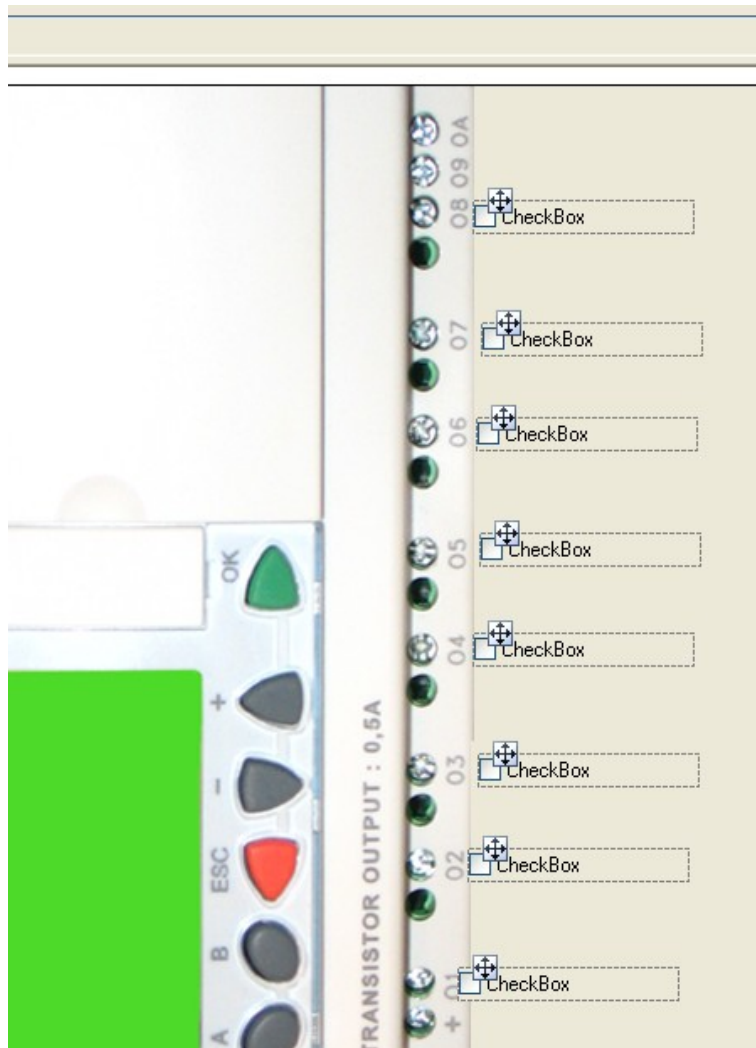
- **Le bouton:** il affiche un texte. Chaque appuis sur le bouton change la valeur de sa donnée associée de 0 à 1 ou de 1 à 0. Il est utilisé pour envoyer des commandes impulsionsnelles à l'automate.
- **La checkbox:** elle affiche un texte. Chaque appuis sur la checkbox change la valeur de sa donnée associée de 0 à 1 ou de 1 à 0. La case est cochée lorsque la donnée vaut 1 et décochée lorsque la donnée vaut 0. Il est utilisé pour envoyer des commandes stables à l'automate.
- **Le slider:** il n'affiche pas de texte. Il permet de régler rapidement une valeur numérique sans affichage de la valeur. Les bornes minimum et maximum du slider sont automatiquement assignées en fonction des bornes de sa donnée associée.
- **L'édit numérique:** il affiche la valeur de la donnée associée. Il permet de saisir au clavier ou avec les flèches « haut » et « bas » une valeur numérique avec affichage de la valeur. Les bornes minimum et maximum de l'édit numérique sont automatiquement assignées en fonction des bornes de sa donnée associée.
- **Le texte:** il affiche un texte. Il ne peut pas avoir de donnée associée.
- **La combobox (liste déroulante):** elle affiche une liste de choix. Elle permet de choisir une valeur dans une liste. La valeur de la donnée associée correspond à l'index du choix: 0 pour le premier choix, 1 pour le second, 2 pour le troisième, etc. La liste des valeurs possibles doit être saisie en tant que texte (voir un peu plus loin), et chaque valeur doit être séparé par un point virgule.


Dans le cadre du tutoriel, nous n'allons pas utiliser de boutons, donc sélectionnez celui posé précédemment sur la surface de dessin, puis supprimez le en utilisant la touche « suppr » du clavier.

Pendant la configuration des trames, nous avons créé des trames pour lire et écrire les registres SL (ou TCP modbus) câblés dans le programme du M3. Notre but est de piloter les sorties TOR de l'automate par état. Nous allons donc utiliser des checkbox.

Ajoutez une checkbox sur la surface de dessin pour chaque sortie du contrôleur M3. Vous pouvez tous les sélectionner en réalisant un clique gauche de la souris sur une zone vide de la surface de dessin, puis en ajustant votre cadre de sélection afin que les contrôles que vous souhaitez sélectionner soient au moins à cheval sur le cadre. Vous pouvez également utiliser la touche « Maj » du clavier et sélectionner les contrôles un par un en cliquant dessus.

ASTUCE: les contrôles que vous ajoutez depuis la page d'outils ont des paramètres par défaut. Si vous souhaitez créer une copie d'un des contrôles que vous avez déjà placé et paramétrés dans la surface de dessin, il vous suffit de maintenir la touche « ctrl » du clavier enfoncé, puis de faire un glisser / déposer en partant du contrôle dont vous souhaitez créer une copie. Ceci aura pour effet de créer un nouveau contrôle dont les paramètres (donnée associée, texte, etc..) seront identiques au contrôle d'origine.



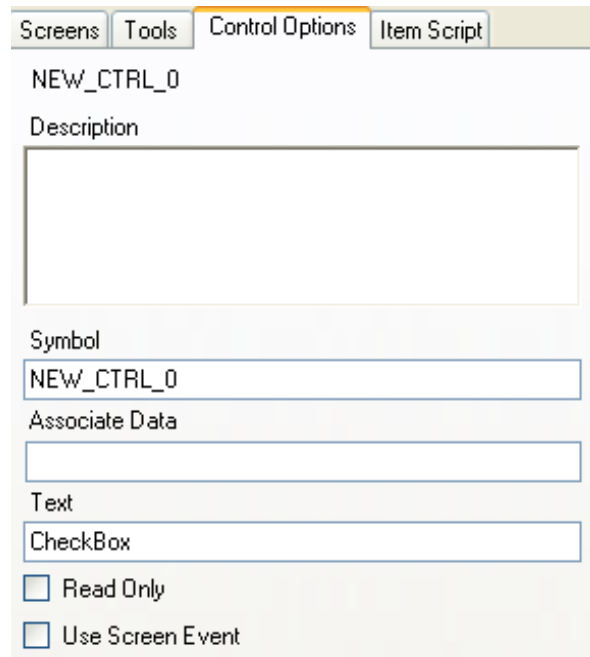
Une fois les contrôles sélectionnés, cliquez sur le bouton « Arrange left » (aligner à gauche) 

Vos contrôles seront alors tous alignés sur la gauche du premier contrôle de la sélection. Lorsque vous avez une sélection multiple, il est également possible de les déplacer tous ensemble en utilisant le bouton de déplacement d'un des contrôles.



Double cliquez maintenant sur une des checkbox ajoutées précédemment. La page « Control Option » s'affiche.

5.3 Control Options Page



Screens Tools **Control Options** Item Script

NEW_CTRL_0

Description

Symbol

NEW_CTRL_0

Associate Data

Text

CheckBox

☐ Read Only

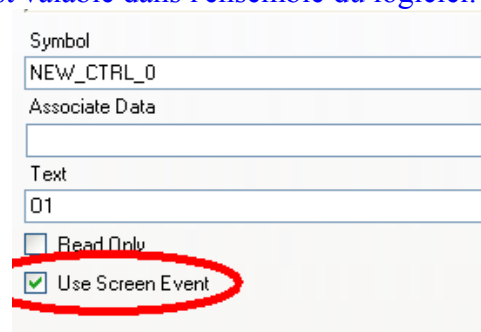
☐ Use Screen Event

Dans cette page, vous trouverez les paramètres des contrôles. .

- **Description**: description du contrôle.
- **Symbol**: symbole du contrôle.
- **Associate Data**: Symbole de la donnée associée qui sera utilisé par le contrôle en mode « Command ». le principe des données associées a été décrit précédemment
- **Text**: le texte affiché par le contrôle.
- **Use Screen Event**: détermine si le contrôle déclenche l'évènement écran et donc exécute le script de l'évènement écran lorsqu'il est actionné. Cette partie sera détaillée un peu plus loin.

À cette étape, vous devez changer le texte de chacune des checkbox ajouté plus tôt dans le script afin de le faire correspondre au nom de la sortie du contrôleur situé juste en face, et cocher le paramètre « Use screen event » pour chacune d'entre elle comme montré dans les captures d'écran suivantes.

NOTE : les modification apportées à un contrôle sont appliqués lorsque vous quitter la page de propriété des contrôles. Par exemple vous pouvez cliquer dans la surface de dessin. Ce principe de validation est valable dans l'ensemble du logiciel.



Symbol

NEW_CTRL_0

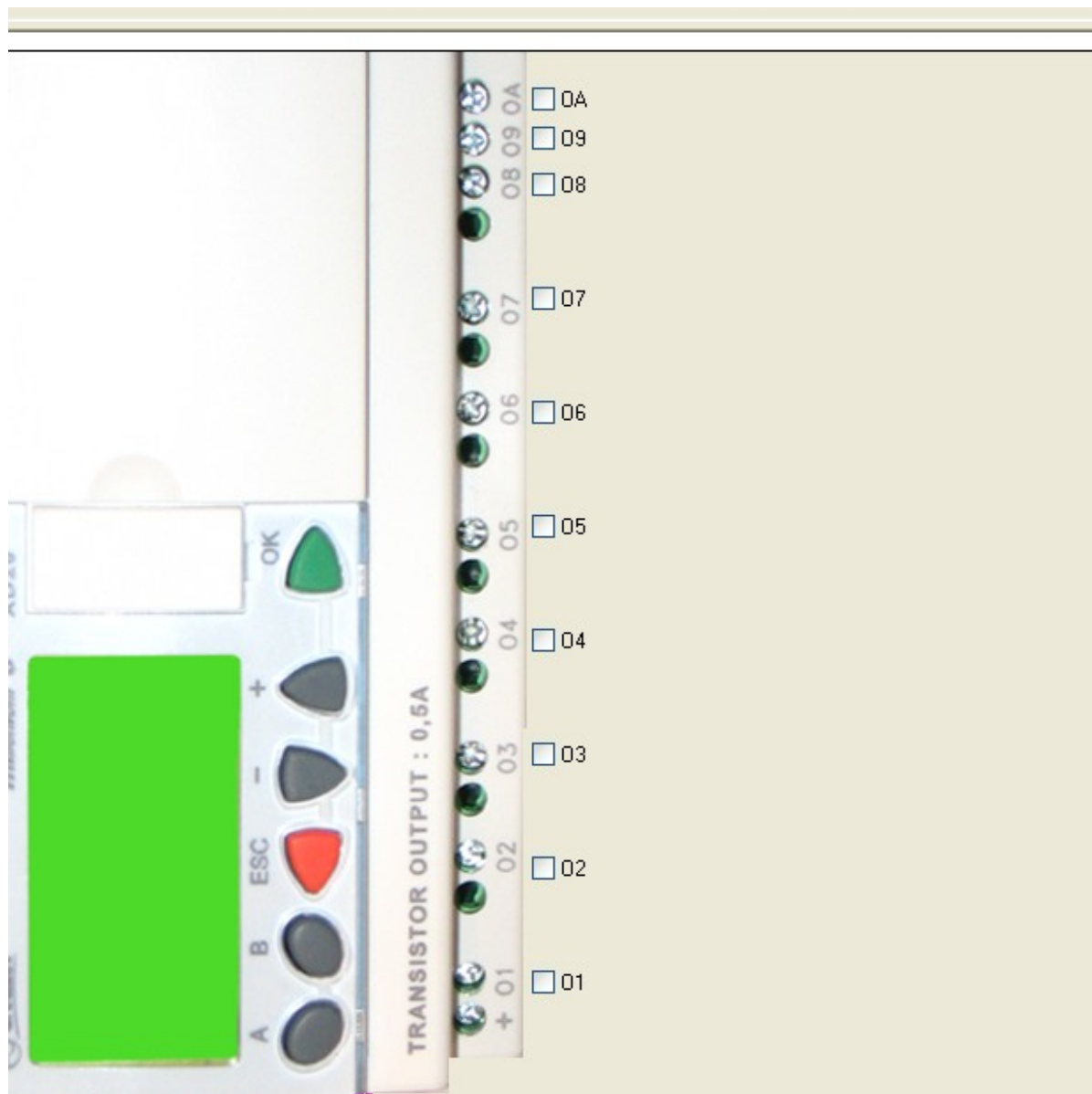
Associate Data

Text

01

☐ Read Only

☒ Use Screen Event



Associer les données et les contrôles.

Comme expliqué précédemment le principe de donnée associé tiens une place prépondérante dans le fonctionnement du logiciel, car c'est ce système qui vous permet de choisir les valeurs envoyées au contrôleur, ainsi que d'afficher les valeurs lues dans ce dernier.

Dans ce tutoriel, nous voulons faire en sorte que chacune checkbox contrôle une sortie du contrôleur. Quand les trames ont été créées, les données nécessaires pour chaque bit (sortie TOR) des blocs DEC/BIN ont aussi été générées. Grâce au câblage réalisé dans le programme du M3, chaque donnée 1 bit crée correspond à une sortie du contrôleur.

Voici une correspondance entre les données et les sorties du contrôleur:

Cas du Wizard TCP modbus:

TCPMB_IN_REG_12_B1: sortie O1 du contrôleur

TCPMB_IN_REG_12_B2: sortie O2 du contrôleur

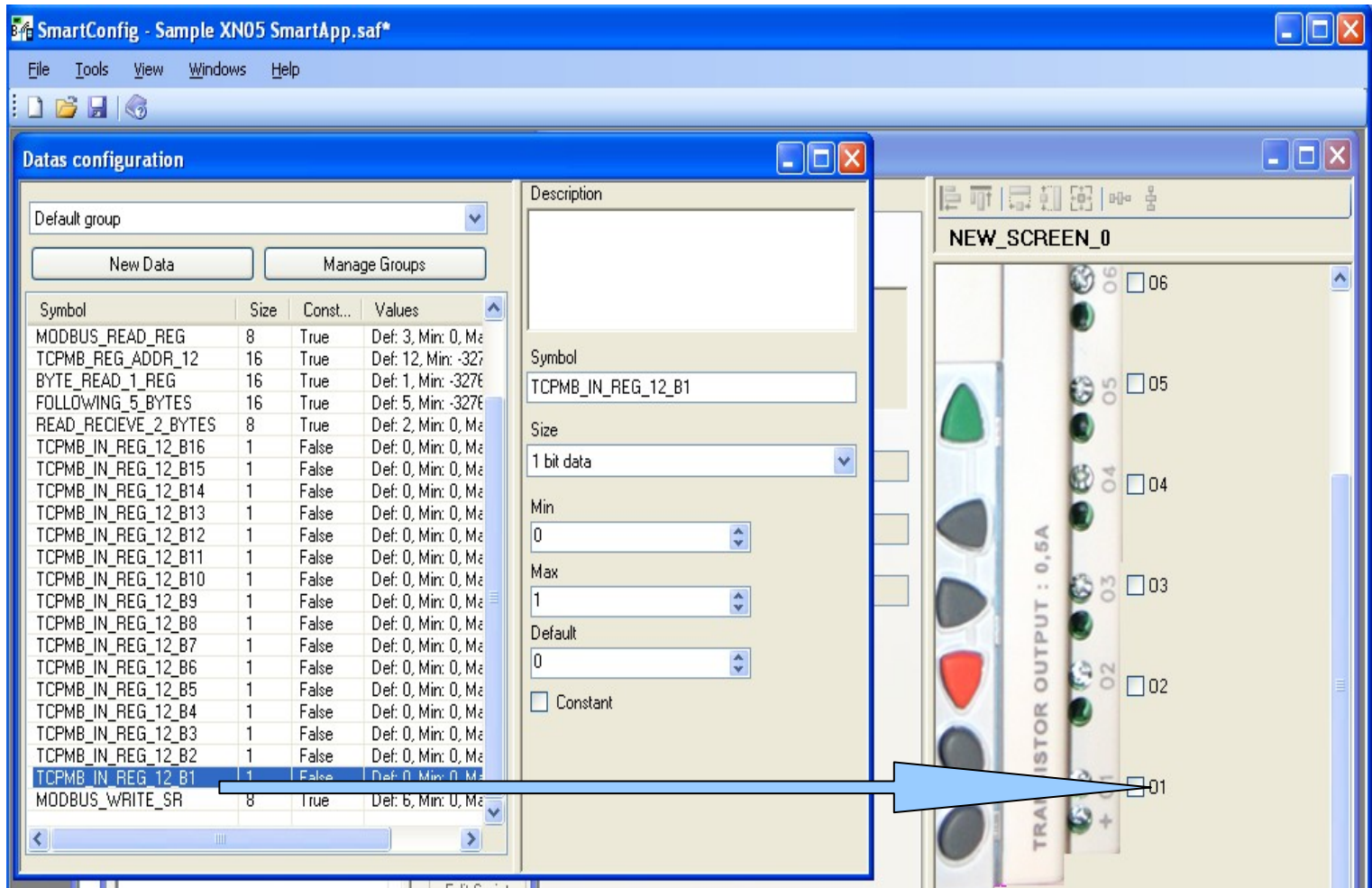
TCPMB_IN_REG_12_B3: sortie O3 du contrôleur
TCPMB_IN_REG_12_B4: sortie O4 du contrôleur
TCPMB_IN_REG_12_B5: sortie O5 du contrôleur
TCPMB_IN_REG_12_B6: sortie O6 du contrôleur
TCPMB_IN_REG_12_B7: sortie O7 du contrôleur
TCPMB_IN_REG_12_B8: sortie O8 du contrôleur
TCPMB_IN_REG_12_B9: sortie O9 du contrôleur
TCPMB_IN_REG_12_B10: sortie OA du contrôleur

M3 SL Bloc Wizard:

M3_SLIN_O1_B1: sortie O1 du contrôleur
M3_SLIN_O1_B2: sortie O2 du contrôleur
M3_SLIN_O1_B3: sortie O3 du contrôleur
M3_SLIN_O1_B4: sortie O4 du contrôleur
M3_SLIN_O1_B5: sortie O5 du contrôleur
M3_SLIN_O1_B6: sortie O6 du contrôleur
M3_SLIN_O1_B7: sortie O7 du contrôleur
M3_SLIN_O1_B8: sortie O8 du contrôleur
M3_SLIN_O1_B9: sortie O9 du contrôleur
M3_SLIN_O1_B10: sortie OA du contrôleur

Arrangez les fenêtre des donnée et la fenêtre de configuration des écrans comme présenté dans la capture d'écran suivante.

Effectuez un glisser/déposer de la fenêtre des données vers le contrôle auquel vous souhaitez ajouter la donnée associée. Pour la checkbox O1, il faudra pendre la donné TCPMB_IN_REG_12_B1 pour le TCP modbus ou la donné M3_SLIN1_O1_B1 pour les blocs SL.



Quand le curseur de la souris autorise le dépôt, relâche le bouton de la souris. Une étoile verte apparaît en haut à droite de la checkbox. Ceci signifie que le contrôle possède une donnée associée (celle que vous venez d'ajouter).

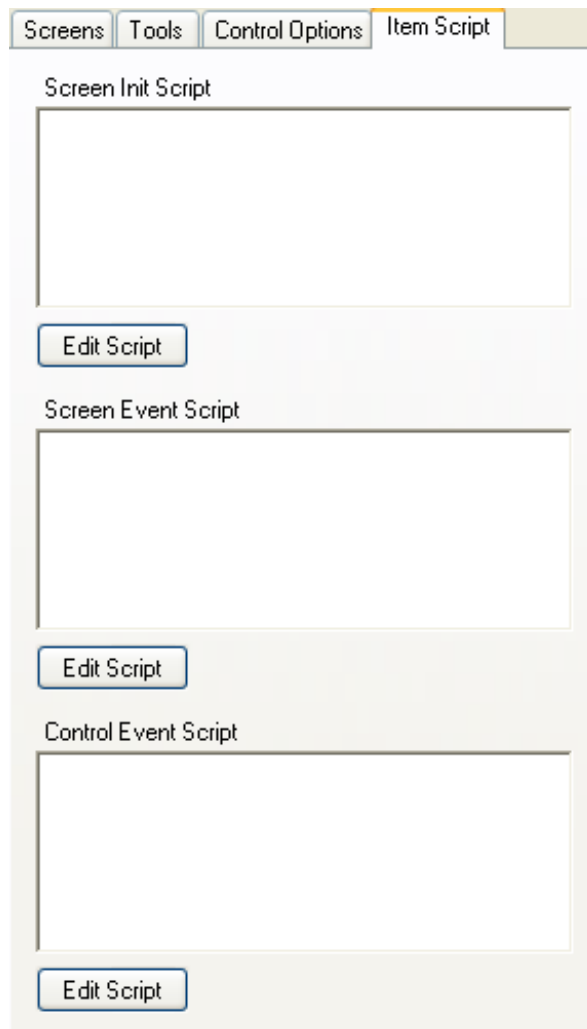
L'association de donnée peut également être faite en saisissant directement au clavier dans le champ « Associate Data » présenté précédemment.

Répétez cette pour chacune des checkbox placée dans la surface de dessin.

Sauvegardez votre fichier et continuez le tutoriel.

5.4 Item Script Page (Page du script des objets)

Activez maintenant la page « Item Script » situé dans la fenêtre de configuration des écrans.



The screenshot shows a software interface for configuring scripts. At the top, there are four tabs: 'Screens', 'Tools', 'Control Options', and 'Item Script'. The 'Item Script' tab is currently selected. Below the tabs, the interface is divided into three main sections, each containing a large text area for script input and a button labeled 'Edit Script'.

- The first section is titled 'Screen Init Script'.
- The second section is titled 'Screen Event Script'.
- The third section is titled 'Control Event Script'.

Dans cette page, vous apercevez 3 boîtes de textes. Ces boîtes de textes contiennent le script qui sera exécuté à différents moment de la vie de l'application en mode « Command ».

NOTE: Le système de script de l'application est très important, il vous permet notamment de définir le moment des trames seront envoyées afin d'écrire ou de lire dans le contrôleur. Quoiqu'il en soit, ne soyez pas impressionné à l'idée d'utiliser un script, celui ci à été conçu de sorte qu'il soit simple à comprendre et à utiliser, dans le but que des non informaticiens puissent l'utiliser après un court temps d'apprentissage. Les possibilités offertes par le script sont définies un peu plus loin.

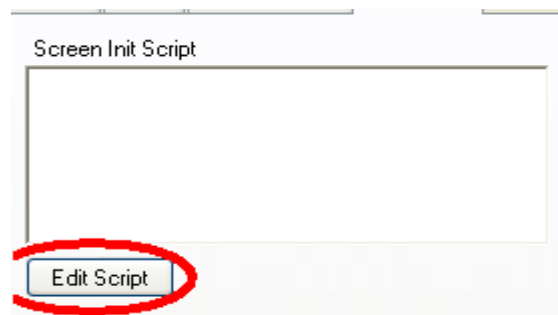
Screen Init Script : Ce script est exécuté au moment où la fenêtre devient active dans le mode « command ». Comme son nom l'indique, ce script a pour but d'initialiser les valeurs des données associées qui sont utilisées par les contrôles ajoutés dans la fenêtre afin d'afficher un état cohérent avec celui des blocs SL ou modbus de l'automate. Ce script a donc une grande importance. De manière générale, il contiendra le script qui ira lire les blocs SL IN (ou entrée modbus) du contrôleur.

Screen Event Script : Ce script est exécuté au moment où un des contrôles pour lequel vous avez coché l'option « Use Screen event script » est actionné en mode « Command ». Comme son nom l'indique, ce script est générale à toute la fenêtre. Son Principale intérêt est d'éviter d'avoir à réécrire pour chaque contrôle un script qui sera plutôt de standard (comme par exemple l'envoi d'une requête d'écriture de 1 ou plusieurs registres / blocs SL)

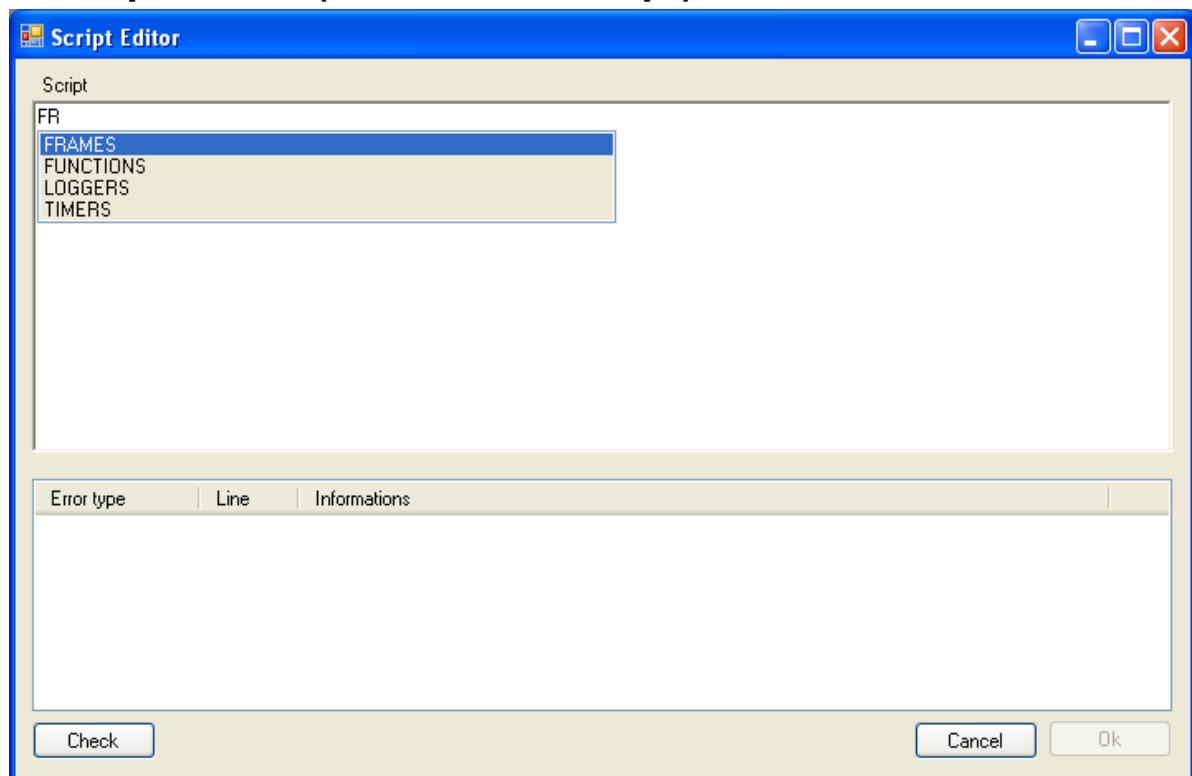
Screen Event Script : Ce script est exécuté au moment où le contrôle couramment sélectionné est actionné en mode « Command ». Comme son nom l'indique, ce script est spécifique à un seul et unique contrôle. Il permet d'exécuter un script spécifique à un seul contrôle comme par exemple démarrer ou arrêter un logger (voir plus loin).

Toutes les trames nécessaires au bon fonctionnement de la supervision ont déjà été créées au début du tutoriel en utilisant les wizards. Voyons maintenant comment les utiliser concrètement via le script.

Cliquez sur le bouton “Edit Script” en dessous de la boîte de texte intitulé « Screen Init script »



6. Script Editor (l'éditeur de script)



Cette fenêtre va vous aider à écrire votre script. De manière générale elle vous proposera spontanément ce que vous avez le droit d'écrire. Lorsque vous saisissez du texte dans la partie supérieur, une liste apparaît vous indiquant et vous permettant de sélectionner directement le script que vous pouvez écrire.

La partie basse de la fenêtre contiendra les éventuelles erreur de script. Cette liste est rafraichie chaque fois que vous cliquez sur le bouton « Check ». Elle indique le type d'erreur, la ligne de l'erreur, ainsi qu'une information précisant le problème.

Pour le moment, le script possède peu de possibilités, mais malgré cette faiblesse, il est très simple à comprendre.

La plus part du temps, une ligne peut être décomposée en 3 éléments séparés par des points, plus une une parenthèse ouvrante ainsi qu'une fermante pouvant contenir des arguments.

- Le premier élément est le type d'objet
- Le second est le nom de l'objet (symbole utilisé partout dans l'application)
- Le troisième est la fonction à appliquer sur l'objet.

Afin de mieux comprend à quoi cela correspond, nous allons étudier un exemple courant: l'envoi d'une trame:

Voici la ligne de script qui permet d'envoyer une trame.

FRAMES . TCPMB_READ_1_IN_REG_FROM_AD12 . SEND()

Dans cette ligne, le premier élément est « FRAMES », ce mot signifie que la ligne va exécuter une instruction en rapport avec les trames.

Le second élément et le symbole de la trame «

TCPMB_READ_1_IN_REG_FROM_AD12 ». Cela signifie que l'instruction va être executée sur la trame ayant ce symbole. Dans notre cas, ce symbole est le symbole par défaut d'une trame généré par le Wizard TCP modbus.

Le dernier élément est la fonction qui sera exécutée suivie de parenthèses « SEND() »

La ligne entière signifie dont que la trame nommée

TCPMB_READ_1_IN_REG_FROM_AD12 sera envoyé au contrôleur lorsque le script sera exécuté.

Type d'objets:

- **Trames:**
 - **Mot clé:** FRAMES
 - **Description:** Les trames sont les messages échangés avec l'automate afin de lire et écrire des valeurs.
 - **Fonction disponibles:**
 - SEND(): Envoie une trame
 - RECIEVE(): Attend la réception (pendant maximum 5 secondes) d'une trame et effectue son décryptage une fois reçue.

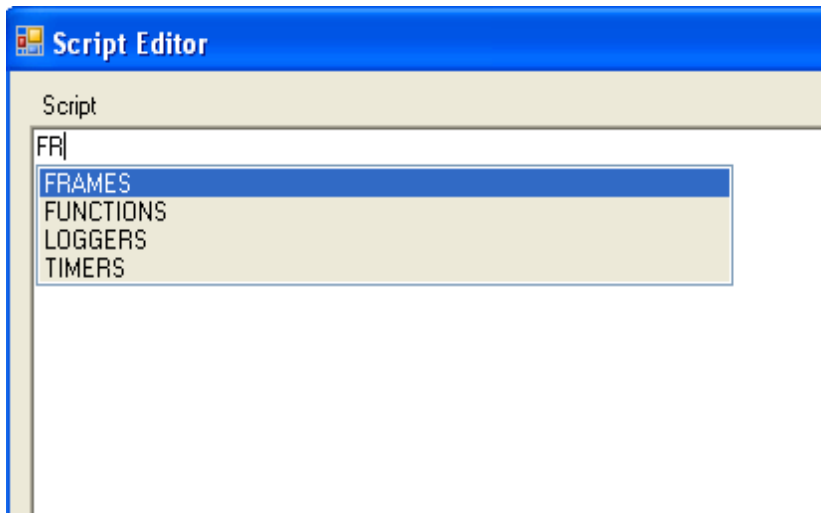
Comme il à été expliqué au début du tutoriel, les trames vont de paire (la requête et la réponse). Donc, lorsque vous envoyer une trame via du script, il est **impératif** d'en recevoir une juste après.

- **Fonctions:**

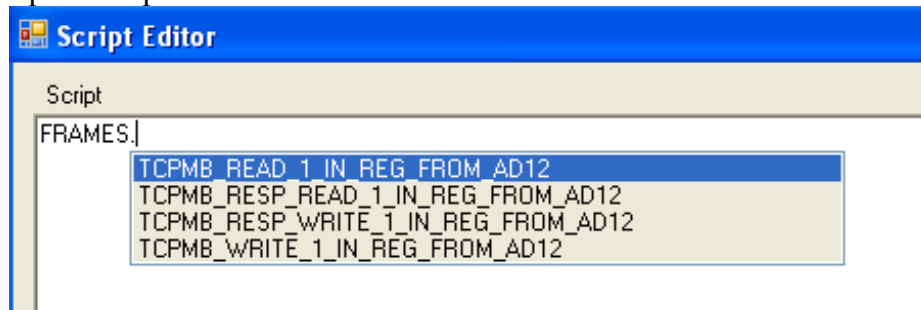
- **Mot clé:** FUNCTIONS
- **Description:** une fonction est un objet qui ne contiennent que du script. Le script qu'elle contiens sera exécuté lorsque la fonction sera appelé dans un script. Le but principale des fonction est de regrouper un script de plusieurs lignes à un seul endroit, puis de l'appeler avec une seule et unique ligne depuis n'importe quel autre script contenu dans l'application. Un exemple courant consiste à crée une fonction qui contiendra l'envoie et la réception de plusieurs trames, et de l'utiliser dans les différents événement qui peuvent être déclenché dans un écran (un seul à été décrit pour le moment, les autres viendront par la suite).
- **Loggers:**
 - **Mot clé:** LOGGERS
 - **Description:** un logger vous permet d'écrire un fichier la valeur courante de certaines données dans un fichier « .csv ». Ce type de fichier peut ensuite être utilisé dans les logiciel comme Excel© afin de tracer des graphiques par exemple. Les précision sur les logger seront données plus loin.
 - **Fonction disponibles:**
 - START(): Démarre le logger automatique
 - STOP(): Arrete le logger automatique
 - LOG(): Ecrit les valeurs des données gérées par le logger dans le fichier
 - CLEAR(): Vide le fichier de log.
- **Timers**
 - **Mot clé:** TIMERS
 - **Description:** Les timers sont utiliser pour executer du script de façon périodique. L'utilisation la plus **importante** et intéressante de cet objet est d'exécuter périodiquement des instructions de lecture des registres/bloc SL de sortie du contrôleur distant afin de rafraichir la supervision.
 - **Fonction disponibles:**
 - START(): Démarre le timer
 - STOP(): Arrête le timer
- **Maths**
 - **Mot clé :** MATH
 - **Description:** Ce mot clé regroupe les fonction mathématiques de base (addition, soustraction, multiplication et division). Elle permettent de faire des opération sur les données utilisés dans votre supervision.
 - **Fonction disponibles:**
 - ADD(résultat, opérateur 1, opérateur 2) : assigne le résultat de l'addition des données « opérateur 1 + opérateur 2 » dans la donnée « résultat ».
 - SUB(résultat, opérateur 1, opérateur 2) : assigne le résultat de la soustraction des données « opérateur 1 - opérateur 2 » dans la donnée « résultat »
 - MUL(résultat, opérateur 1, opérateur 2) : assigne le résultat de la multiplication des données « opérateur 1 x opérateur 2 » dans la donnée « résultat »
 - DIV(résultat, opérateur 1, opérateur 2) : assigne le résultat de la division des données « opérateur 1 / opérateur 2 » dans la donnée « résultat »

Revenons à notre éditeur de script:

Commencez à écrire le mot clé « FRAMES » dans l'éditeur de script. Une liste de choix apparaît.



Dans cette liste, vous trouverez tout les objets autorisés à cet endroit du script (en l'occurrence ce sont les mots clé vu précédemment). En appuyant sur la touche « flèche bas » du clavier, vous activez la liste d'auto complétion. Une fois cette liste activé , en appuyant sur les touches « flèche haut » et flèche bas » vous pouvez sélectionner le mot clé qui vous intéresse. Sélectionnez le mot « FRAMES » et appuyez sur la touche « Entrée » sur clavier. Le texte « FRAMES » à été écrit automatiquement dans l'éditeur de script. Maintenant, ajoutez un point. Une fois de plus la liste d'auto complétion s'affiche et vous propose ce que vous pouvez saisir mais avec cette fois ci une liste de choix différente.



Étant donné que le premier mot est « FRAMES » les choix qui vous sont proposés sont les symboles des différentes trames présentes dans votre fichier de configuration.(ici, c'est la liste des trames générés par le Wizard TCP modbus). Quelque soit le protocole utilisé et comme expliqué précédemment, un échange complet avec un contrôleur distant se fait avec deux trames (requête et réponse). Dans notre cas, nous souhaitons lire les registres /Bloc SL du contrôleur afin d'initialiser les données aux valeurs contenues dans le contrôleur. Nous allons donc envoyer une trame pour demander les valeurs du contrôleur puis réceptionner la réponse du contrôleur.

Le script à écrire sera donc :

Cas du TCP modbus

Nous envoyons la trame de requête de lecture de 1 registre à partir de l'adresse 12

FRAMES.TCPMB_READ_1_IN_REG_FROM_AD12.SEND()

Puis nous recevons la trame de réponse

FRAMES.TCPMB_READ_1_IN_REG_FROM_AD12_RET.RECEIVE()

Cas de la liaison série M3 (bloc SL)

Nous envoyons la trame de requête de lecture du bloc SL IN 1 (1-8)

FRAMES.M3_READ_SLIN_BLOC_1.SEND()

Puis nous recevons la trame de réponse

FRAMES.M3_READ_SLIN_BLOC_1_RET.RECEIVE()

une fois que le script est écrit, cliquez sur le bouton « Check script ». Si il n'y a pas d'erreur, le bouton « OK » devient disponible. Sinon la liste des erreur vous indique les problèmes et le bouton reste grisé.

Cliquez sur le bouton OK puis ouvrez l'éditeur de script pour le « Screen Event Script »

Lorsque vous avez ajouté et configuré les les checkbox, vous avez coché l'option « Use Screen Event Script », et c'est ici que cela va devenir utile.

Étant donné que les checkbox sont censé piloter les sorties de l'automate, il faut que lorsque qu'elles sont actionné, elles envoient leurs valeurs vers l'automate. Il faut donc qu'une action sur ces dernière déclenche un envoie de trame d'écriture.

En utilisant le mot clé « FRAMES », de la même manière que pour la lecture, nous allons d'abord envoyé la requête d'écriture et ensuite lire la réponse.

Le script sera donc:

Cas du TCP modbus

FRAMES.TCPMB_WRITE_1_IN_REG_FROM_AD12.SEND()

FRAMES.TCPMB_WRITE_1_IN_REG_FROM_AD12_RET.RECEIVE()

Cas de la liaison série M3 (bloc SL)

FRAMES.M3_WRITE_SLIN_BLOC_1.SEND()

FRAMES.M3_WRITE_SLIN_BLOC_1_RET.RECEIVE()

une fois le script écrit, sauvegardez votre fichier.

Récapitulatif:

Qu'avons nous fait depuis le début de ce tutoriel (car pour l'instant, vous n'avez pas fait grand chose qui communique avec le M3...je sais, c'est frustrant):

- Nous avons crée des trames
- Parcouru la fenêtre des trames et des données.
- Crée une fenêtre qui représente le M3 et permet de piloter ses sorties
- Écrit 4 lignes de script (deux pour lire les blocs SL IN / registres d'entrée, et deux pour les écrire)

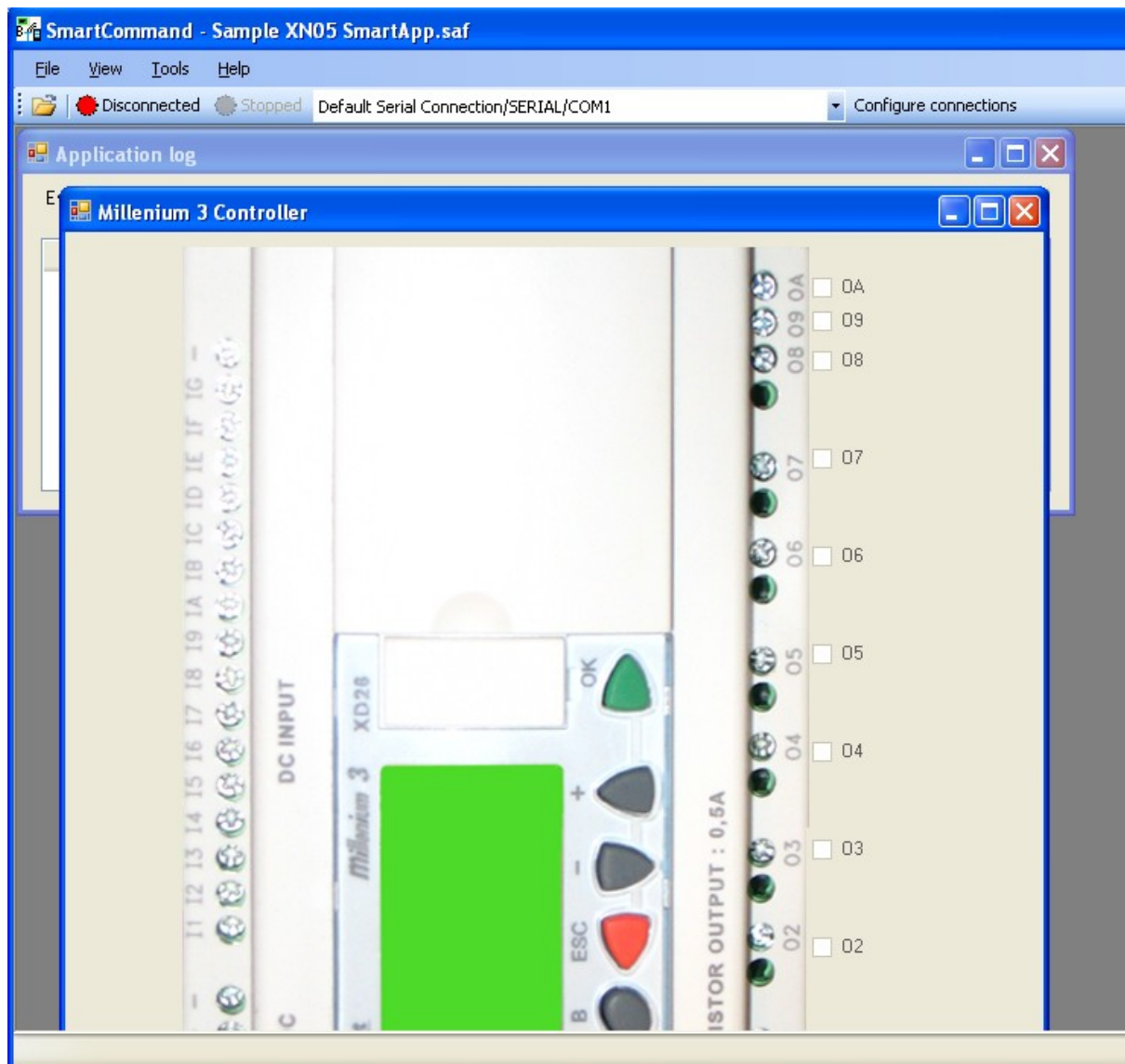
Oui je sais, résumé comme ça, on à l'impression d'avoir rien fait, mais détrompez vous, car avec « si peu », vous allez maintenant piloter les sorties de votre M3 depuis votre PC (si si)

Via le Menu « Tool » cliquez sur la commande « Jump to Smart Command » ou pressez simplement la touche F5 de votre clavier.

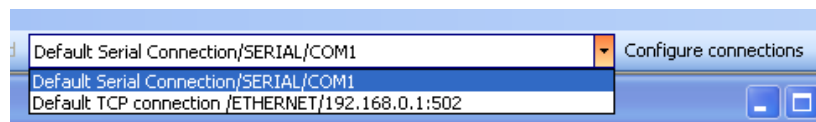
L'application Smart Command s'ouvre en chargeant automatiquement votre fichier et Smart Config disparaît.

7. SmartCommand

Le but de cette application est d'exécuter tout ce que vous avez configuré jusqu'ici dans votre fichier de supervision, en connexion directe avec votre contrôleur distant. Voici à quoi ressemble Smart Command une fois votre fichier chargé.

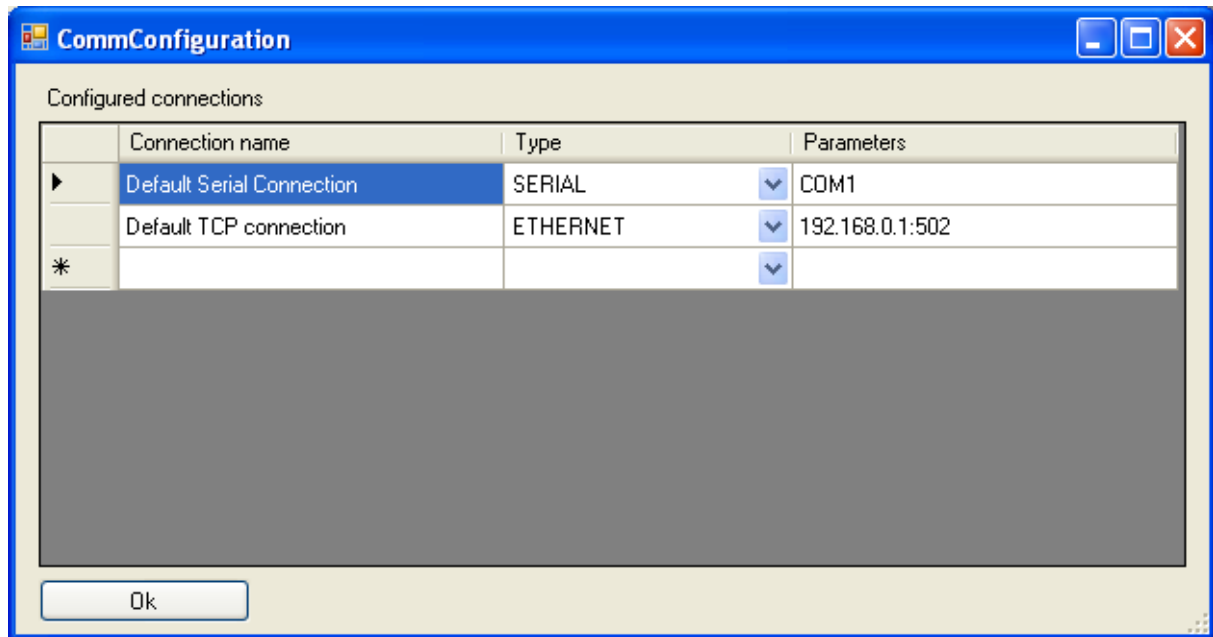


7.1 Connection configuration (configuration de la connexion)



Dans la barre d'outils, vous trouverez une liste des connexions disponibles ainsi qu'un bouton « Configure connections ». Par défaut, deux connexion différentes sont disponibles (voir la capture d'écran)

En cliquant sur le bouton « Configure connections », la fenêtre suivante apparaît.



Afin de dialoguer avec votre automate, vous devez créer une nouvelle connexion ou modifier une connexion existante.

Dans la première colonne, vous pouvez donner un nom à votre connexion. Dans la seconde colonne, vous pouvez choisir le type de connexion en fonction de votre besoin.

Les types de connexion disponibles sont:

- **Ethernet** : le protocole TCP modbus passe par une connexion Ethernet, c'est celle-ci que vous devrez utiliser dans le cas où vous utilisez l'extension XN05.
- **Serial** : ici, pas de surprise, c'est une connexion série standard, c'est celle-ci que vous utiliserez pour les blocs SL:
- **Virtual** : La connexion virtuelle vous permet de tester votre configuration sans avoir de contrôleur sur lequel vous connecter. Son fonctionnement ne sera pas détaillé dans cette version du tutoriel. NB: la connexion virtuelle ne possède aucune intelligence, elle vous permettra simplement de tester que vous écrivez bien les bonnes valeurs dans les bonnes données, et de contrôler que les valeurs que vous lisez fonctionnent bien et réagissent aux différents contrôles que vous avez placés dans vos écrans.

Dans la dernière colonne, vous devez saisir les paramètres de votre connexion.

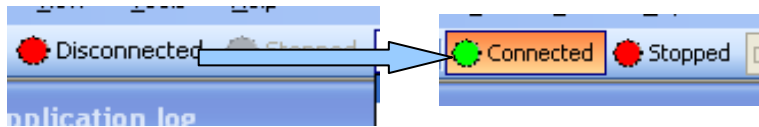
- Pour une connexion série, ce sera le nom du port COM que vous utilisez pour communiquer avec votre contrôleur.
- Pour une connexion Ethernet, ce sera l'adresse IP de l'extension XN05, puis le caractère deux points (:) puis le port de connexion (502 pour le protocole TCP modbus)
- Pour une connexion virtuelle, cette colonne peut rester vide.

Une fois votre connexion configurée, pressez le bouton OK

NOTE: les connexions configurées sont automatiquement sauvegardées et restaurées lors de l'ouverture de Smart Command.

7.2 Démarrer votre supervision.

Sélectionnez votre connexion dans la liste, puis cliquer sur le bouton « Disconnected » de la barre d'outil.



Si vous avez correctement configuré votre connexion, ce bouton doit changer d'état et afficher le texte « Connected », de plus, le bouton voisin « Stopped » devient disponible. Tant que votre supervision n'est pas démarrée, rien ne peut se produire, aucune trame ne peut être envoyée et vous ne pouvez agir sur rien. Quand vous cliquez sur le bouton « Stopped », ce dernier change d'état, affiche le text « Runing » et tout se met à fonctionner.

Bon d'accord, simplement dire « tout se met à fonctionner » est un peu vague, alors voici plus précisément ce qui se passe.

Tout démarrage demande une initialisation... souvenez vous, vous avez écrit un script d'initialisation (Screen Init Script) qui devait lire dans le contrôleur. Bah c'est précisément à ce moment qu'il est exécuté. Ainsi, la fenêtre que vous avez configuré met à jours les données à partir de ce qui est lu dans le contrôleur et vous obtenez de la sorte un état cohérent entre ce que vous affichez, et ce qui est contenu dans le contrôleur.

De la même manière, si d'autre objet comme les timers ou les loggers sont configurés en démarrage automatique, c'est à ce moment qu'ils vont être démarrés. (ces objets seront décrit plus loin).

Vous avez aussi sans doute remarqué que tout ce qui était grisé jusqu'ici dans votre fenêtre s'est dégrisé.

NOTE: Étant donnée que jusqu'ici vous n'aviez rien envoyé comme commande sur votre automate, toutes les checkbox doivent apparaître décochées.

Vous pouvez donc maintenant agir sur vos belles checkbox et piloter ainsi les sorties de votre M3. Si vous avez bien suivi ce tutoriel, chaque fois que vous cliquez sur une des checkbox, la sortie correspondante doit s'actionner (sortie active lorsque la checkbox est cochée). Si vous avez un contrôleur avec des sorties relai, vous devez entendre ces derniers « claquer ». Si vous avez un contrôleur avec des sorties transistor, vous pouvez vérifier qu'elle sont bien passé à l'état actif via la face avant du contrôleur (les numéro des sorties passent en inverse vidéo).

Petit passage déconne:

C'est t'y pas beautifull ça mon ami, tu peux piloter ton M3 à distance depuis ton PC et tout ça grasse à un masochiste de nain formaticien qui à passé des heures et des heures devant son PC pour crée un logiciel qu'il destinait à l'origine à son usage personnel et qui saches le, l'utilise presque quotidiennement chez lui...car dans sa grande fainéantise d'informaticien...même bouger les doigts de sa souris pour appuyer sur un interrupteur représente un effort inutile.

Résumé:

Durant tout le début de ce tutoriel, vous avez compris les bases du principe de supervision, qui consiste à lire et écrire dans le contrôleur. Mais pour le moment, ces opération ne se font

que lorsque vous agissez manuellement sur la supervision (passage en « run » ou action sur les checkbox).

Vous avez sûrement compris que le problème qui se pose, va être de lire automatiquement et périodiquement ce qu'il se passe dans le contrôleur afin que tout se rafraichisse automatiquement. C'est ce que je vous propose d'apprendre dans la suite du tutoriel.

Arrêtez votre supervision via le bouton « Running » et déconnectez l'application via le bouton « Connected », puis fermez SmartCommand. Smart Config reviens exactement comme vous l'avez laissé.

8.Finaliser la Supervision de votre M3

Dans la supervision, le plus intéressante est d'afficher en permanence l'état des entrée et des sorties (ou tout autre valeur interne de votre programme M3)

Comme nous n'avons pas encore créé les trames pour les blocs SL OUT (ou les blocs NUM OUT) nous allons refaire un tour dans les Wizards.

Les paramètres des trames à créer sont les suivants:

Bloc SL M3 :

Frame Type = M3 SL Output Bloc

Read/Write = Read (ne peut être modifié)

Address range = 25-32

M3_SLOUT1_I1 doit être découpé en 16 donnée de 1 bit comme vous l'avez fait pour la sortie du bloc SL IN connecté à un bloc DEC/BIN.

TCP Modbus:

Register Type = Output Register

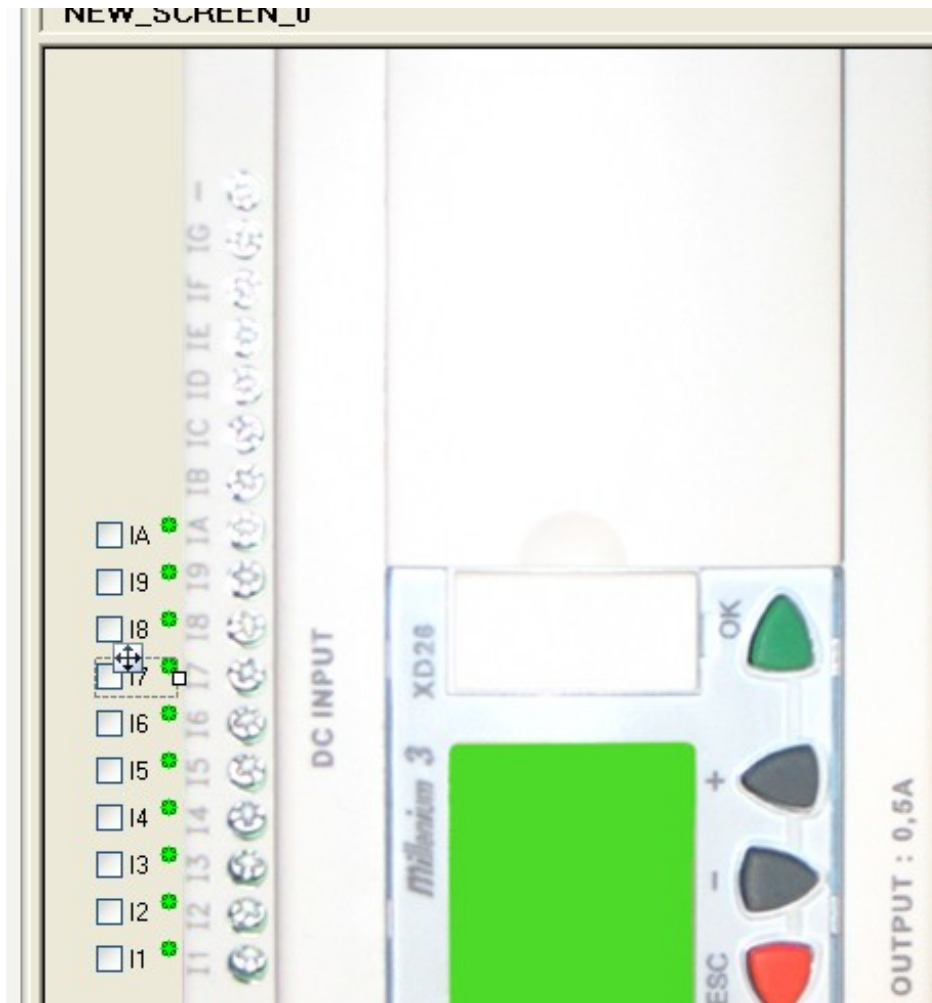
Read/Write = Read Multiple Register.

Starting Address = 20

Number of register = 1

TCPMB_OUT_REG_20 doit être découpé en 16 donnée de 1 bit comme vous l'avez fait pour la sortie du bloc NUM IN connecté à un bloc DEC/BIN.

Maintenant, les trames nécessaire pour lire l'état des entrées du contrôleur ont été créées. Et nous pouvons ajouter de nouveaux contrôles dans notre fenêtre afin d'afficher leurs états. Allez dans la fenêtre « Screen Configuration » et sélectionnez l'écran que vous avez créé précédemment puis ajoutez une checkbox pour chaque entrée TOR du contrôleur (I1 à IA). (voir la capture d'écran suivante).



Voici la liste des association de données à réaliser:.

Wizard TCP modbus:

TCPMB_OUT_REG_20_B1: Entrée I1 du contrôleur
 TCPMB_OUT_REG_20_B2: Entrée I2 du contrôleur
 TCPMB_OUT_REG_20_B3: Entrée I3 du contrôleur
 TCPMB_OUT_REG_20_B4: Entrée I4 du contrôleur
 TCPMB_OUT_REG_20_B5: Entrée I5 du contrôleur
 TCPMB_OUT_REG_20_B6: Entrée I6 du contrôleur
 TCPMB_OUT_REG_20_B7: Entrée I7 du contrôleur
 TCPMB_OUT_REG_20_B8: Entrée I8 du contrôleur
 TCPMB_OUT_REG_20_B9: Entrée I9 du contrôleur
 TCPMB_OUT_REG_20_B10: Entrée IA du contrôleur

M3 SL Bloc Wizard:

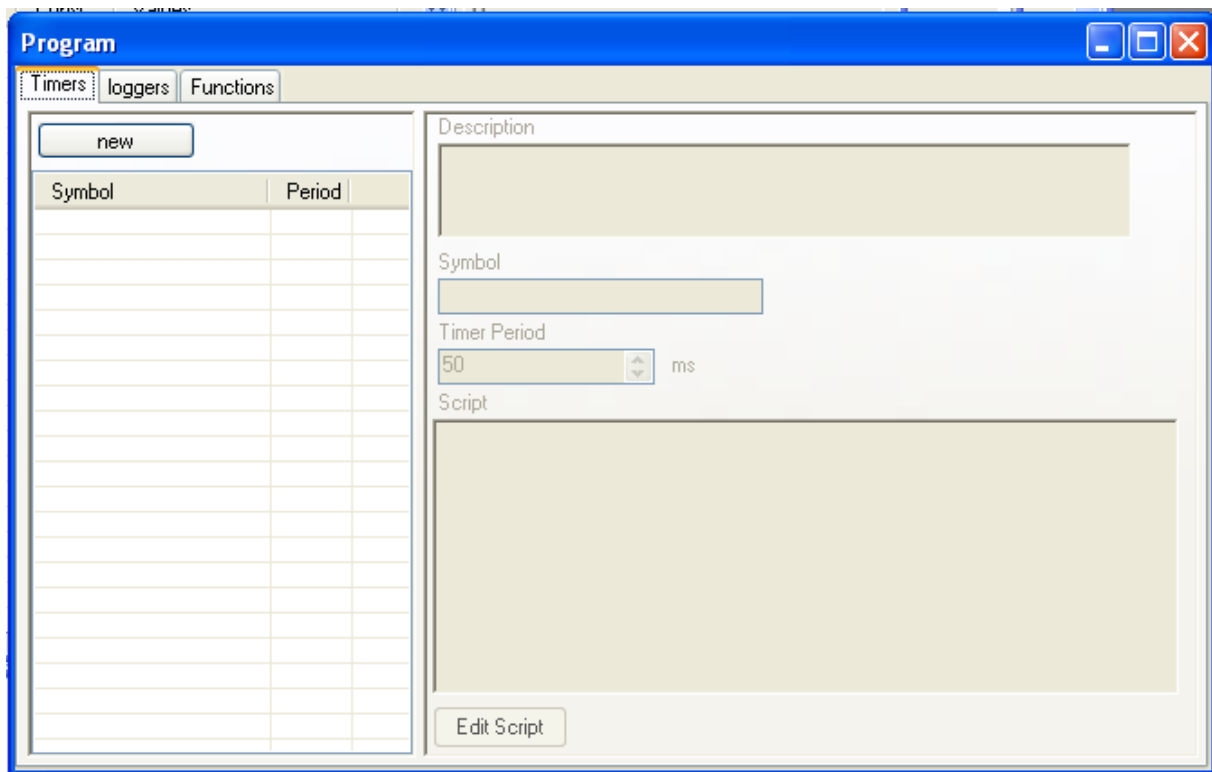
M3_SLOUT_I1_B1: Entrée I1 du contrôleur
 M3_SLOUT_I1_B2: Entrée I2 du contrôleur
 M3_SLOUT_I1_B3: Entrée I3 du contrôleur
 M3_SLOUT_I1_B4: Entrée I4 du contrôleur
 M3_SLOUT_I1_B5: Entrée I5 du contrôleur
 M3_SLOUT_I1_B6: Entrée I6 du contrôleur
 M3_SLOUT_I1_B7: Entrée I7 du contrôleur

M3_SLOUT_I1_B8: Entrée I8 du contrôleur
M3_SLOUT_I1_B9: Entrée I9 du contrôleur
M3_SLOUT_I1_B10: Entrée IA du contrôleur

Une fois que vous avez terminé avec la configuration de votre écran (posé toutes les checkbox, changé leur texte et associé les bonnes données) activez la fenêtre de programmation (« Program Window »)

NB: pour toutes les checkbox que vous venez d'ajouter, elle ne sont pas censé exécuter un quelconque script et donc l'option « Use Screen Event Script » peut rester décochée.

9. Program Window



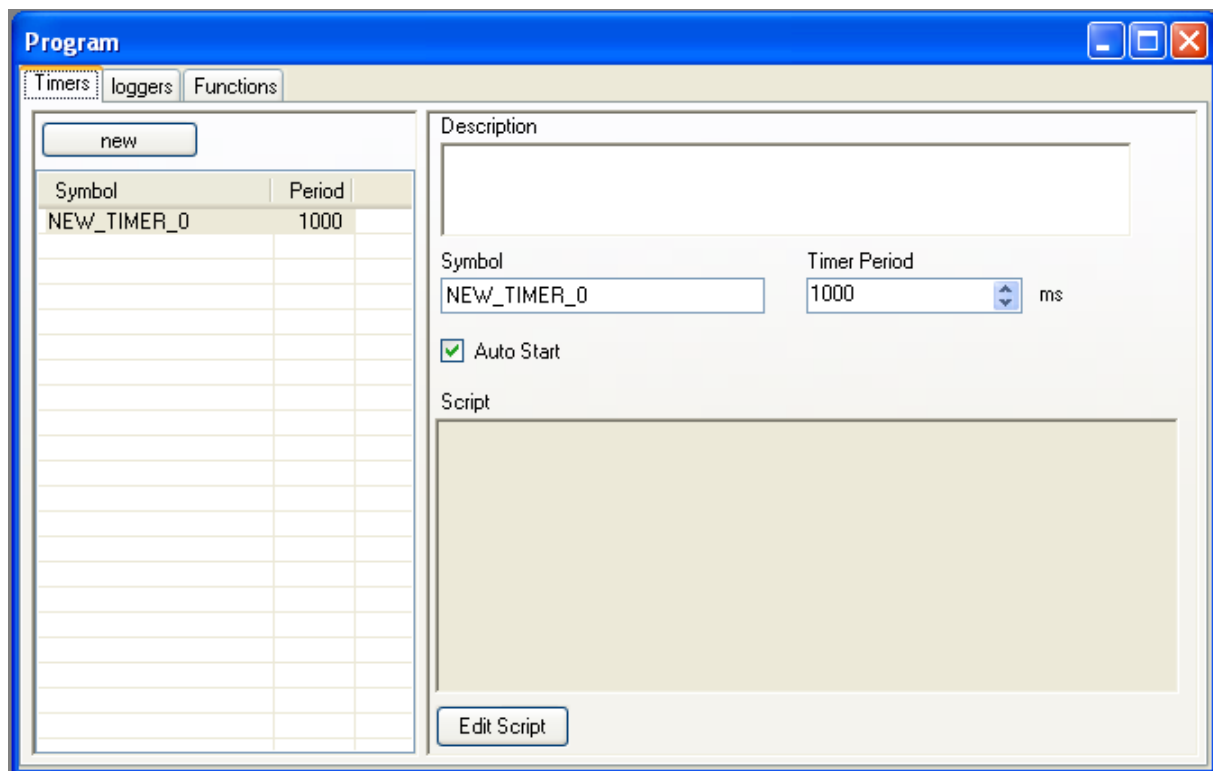
Cette fenêtre se décompose en 3 pages, une pour chaque type d'objets.

La première page est celle des « Timers ».

La seconde est celle des « Loggers »

La dernière est celle des « Functions »

9.1 Timers Page (page des Timers)



Il n'y a pas grand chose à expliquer ici. Un timer est un objet qui exécute périodiquement un script en fonction de sa période (en millisecondes).

Les timer possèdent deux paramètres:

- Period : Ceci représente la période d'exécution en millisecondes.
- Auto Start : Si cette option est cochée, le timer sera automatiquement lancé au passage en run de l'application SmartCommand. Sinon il faudra le démarrer en utilisant du script (vu précédemment, voir les détails des fonctionnalités du script). Un timer qui a l'option « Auto Start cochée » pourra également être arrêté et redémarré par script.

Dans notre exemple, nous voulons lire périodiquement l'état des entrées du contrôleur, il nous faut donc lire le bloc SL out / NUM out.

Le script sera donc:

M3 SL Bloc:

```
FRAMES.M3_READ_SLOUT_BLOC_1.SEND()  
FRAMES.M3_READ_SLOUT_BLOC_1_RET.RECEIVE()
```

TCP Modbus:

```
FRAMES.TCPMB_READ_1_OUT_REG_FROM_AD20.SEND()  
FRAMES.TCPMB_READ_1_OUT_REG_FROM_AD20_RET.RECEIVE()
```

N'oubliez pas de changer la période du script à 100ms afin que la lecture soit assez rapide (la période minimum est de 50 ms).

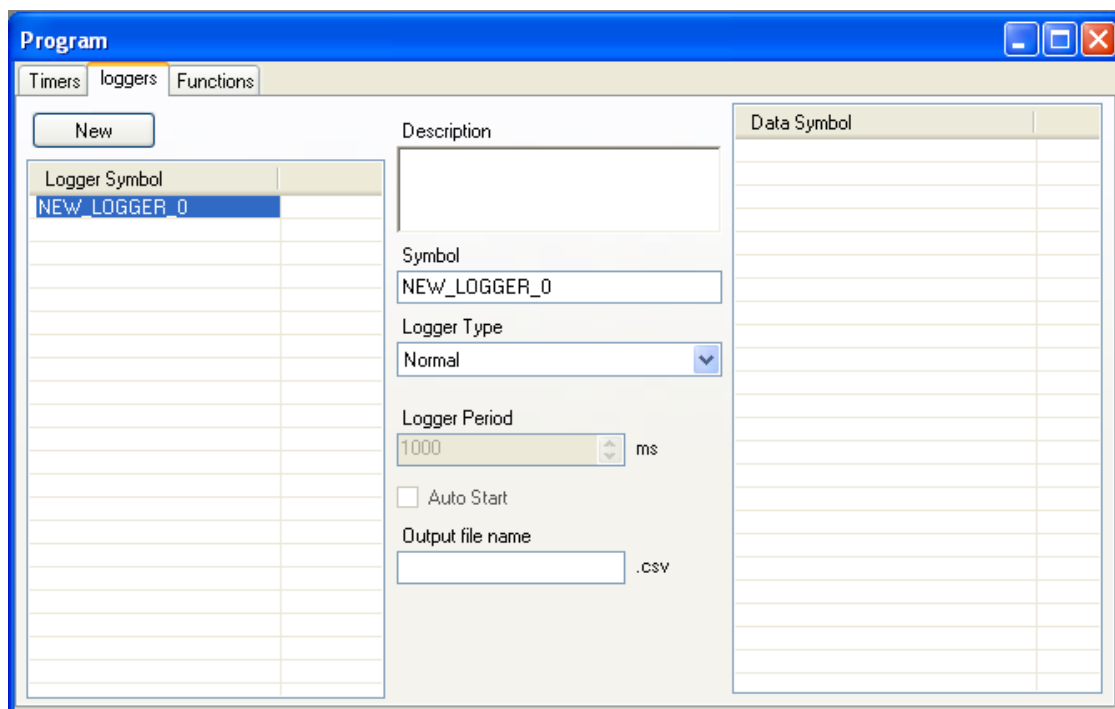
Une fois le script ajouté dans le timer nouvellement créé, vous pouvez sauvegarder votre fichier et retourner dans SmartCommand, vous connecter à votre M3, et passer l'application

en RUN afin que la magie s'opère, vous affichez maintenant en permanence l'état de vos entrée.

Vous en savez maintenant assez pour créer une supervision adaptée à votre besoin, le reste n'est que du bonus vous offrant d'autres fonctionnalités. Une fois que vous aurez bien pris en main cette application, il vous faudra en moyenne 2h pour créer une supervision adaptée à un programme M3 particulier.

9.2 Loggers Page (page des Loggers)

(A venir)

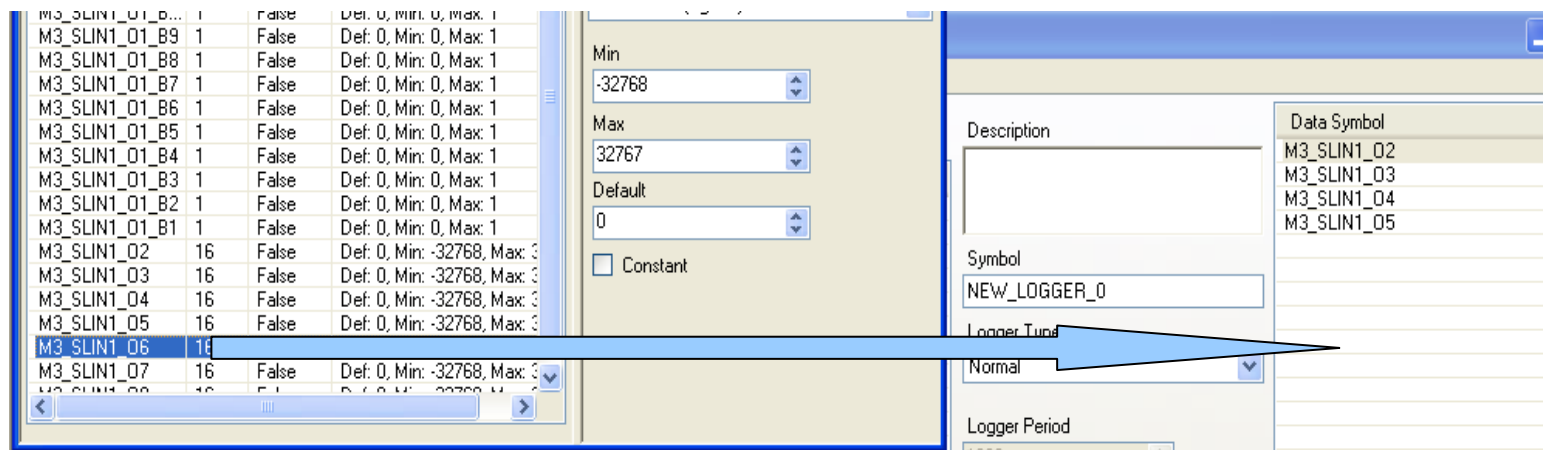


Le Logger permet de stocker les valeurs de données issues de la supervision dans un fichier texte au format CSV (fichier qui peut notamment être importé dans un tableur comme Excel ou Open Office Calc)

Le Logger possède les paramètres suivant:

- **Logger Type:** Il existe deux type de logger: le normal et l'automatique. Un logger normal n'archivera la valeur courante ses données que lors de l'appel à la fonction « LOG » sur le logger par du script. Un logger automatique archivera ses données à la période donnée par la paramètre « Logger Period ». Un Logger automatique peut être arrêté et démarré en utilisant les fonctions « START » et « STOP » du script sur le logger.
- **Logger Period:** période d'archivage d'un logger. Disponible uniquement si le logger est automatique.
- **Auto Start:** Démarrage automatique lors du passage en RUN dans l'application SmartCommand. Disponible uniquement si le logger est automatique.
- **Output file name:** nom du fichier CSV ou seront écrites les valeurs. L'emplacement du fichier est le dossier défini dans les options de SmartCommand.

Pour ajouter une donnée au logger, il suffit de faire un glisser / déposer depuis la fenêtre des données vers la liste des données gérées par le logger.



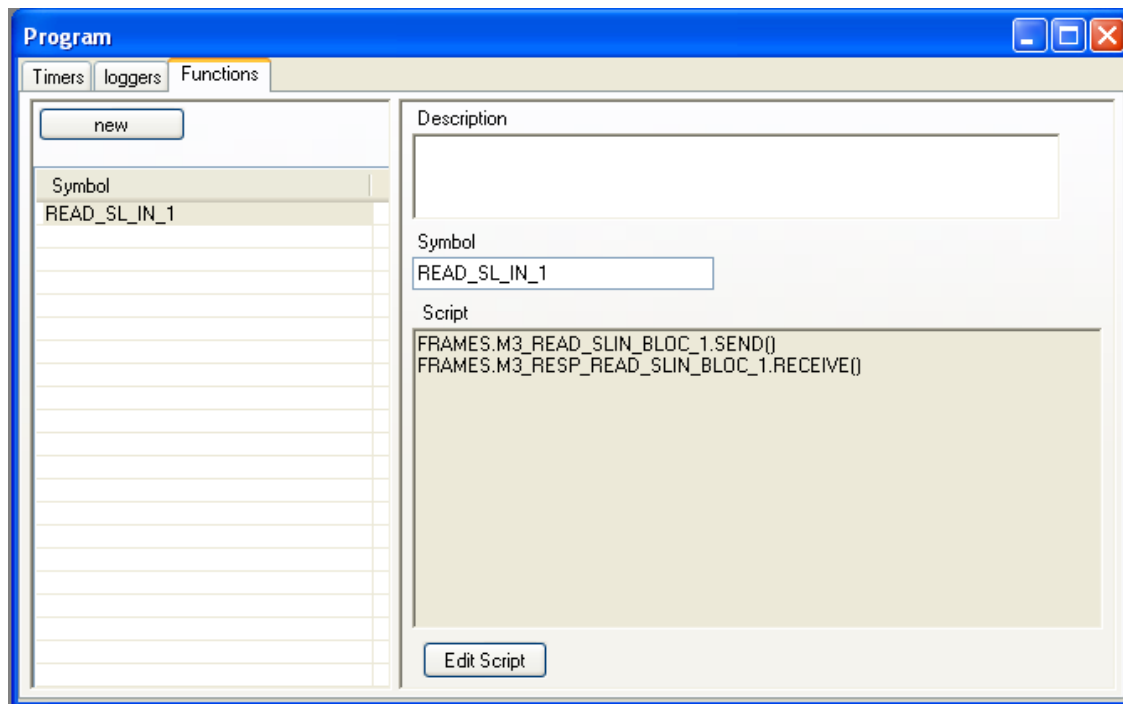
Quelques usages standard du logger:

Dans le cas où vous souhaitez archiver les valeurs de vos données de manière synchrone à un l'exécution d'un timer, il est recommandé de ne pas utiliser un timer automatique. La solution la plus adéquate dans ce cas, et d'appeler la fonction « LOG » sur le logger voulu directement à la suite du script d'envois et réception d'un trame.

Afin de vider votre fichier de log, il convient d'utiliser un bouton qui appellera la fonction « CLEAR » sur le logger dans son script qui lui est propre (« Control Event Script »).

Si vous souhaitez vider votre fichier à chaque démarrage de la supervision, vous pouvez appeler cette fonction dans le script d'initialisation d'une de vos fenêtres (« Screen Init Script »)

9.3 Functions Page (page des fonctions)



Comme expliqué précédemment, une fonction permet d'écrire un script contenant plusieurs instructions, puis de l'appeler dans un autre script en une seule ligne. Dans l'exemple présenté dans la capture d'écran, nous avons une fonction nommée « READ_SL_IN_1 » qui contient le script suivant:

```
FRAMES.M3_READ_SLIN_BLOC_1.SEND()
```

```
FRAMES.M3_RESP_READ_SLIN_BLOC_1.RECEIVE()
```

Ce script permet de lire le bloc SLIN 1 (adresse 1-8) dans le M3.

Si dans n'importe quel autre script vous saisissez la ligne suivante :

« FUNCTIONS.READ_SL_IN_1() », elle aura pour effet d'exécuter les deux lignes contenues dans la fonction.

ATTENTION: une fonction ne doit JAMAIS s'appeler elle-même!!!! (il n'y a pas de contrôle fait par le logiciel dans ce cas particulier, et ceci aura pour effet d'amener l'application SmartCommand à un crash)

Vous l'aurez compris, si votre supervision utilise l'ensemble des blocs SL (ou entrée/sorties d'extension XN05), les fonctions vous permettront de regrouper tout votre script de lecture et d'écriture dans une seule fonction, ce qui peut être pratique pour l'utiliser simplement dans les timers, scripts d'initialisation, ou scripts d'événements.

Avec les calculs mathématiques, (ADD, SUB, MUL, DIV), les fonctions vont aussi vous permettre de grouper une suite d'opérations visant à effectuer un calcul sur une ou plusieurs données.

10 ANNEXES : Les contrôles avancés

Généralité sur les contrôles avancés:

Contrairement aux contrôles standard, ceux ci ne sont pas interne au logiciel, mais ils existent sous forme de DLL que l'on peut ajouter dans l'application sans avoir à réinstaller une nouvelle version. J'ai développé ce système dans le but de pouvoir crée de nouveaux contrôles à la demande et adaptés à chaque type d'application dans lequel le M3 est utilisé (Régulation solaire, Aquariophilie, et tout ce que la communauté d'utilisateur de M3 peut imaginer)

10.1 le rectangle coloré

(A venir)

10.2 l'ellipse coloré

(A venir)

10.3 la jauge

(A venir)

10.4 le serpent

(A venir)

10.5 le contrôle 2 images

(A venir)

10.6 le contrôle 4 images

(A venir)

10.7 l'afficheur formaté

(A venir)

10.8 le ballon d'eau.

(A venir)