

# Modicon M340

Liaison série

Manuel utilisateur

(Traduction du document original anglais)

12/2018

---

Le présent document comprend des descriptions générales et/ou des caractéristiques techniques des produits mentionnés. Il ne peut pas être utilisé pour définir ou déterminer l'adéquation ou la fiabilité de ces produits pour des applications utilisateur spécifiques. Il incombe à chaque utilisateur ou intégrateur de réaliser l'analyse de risques complète et appropriée, l'évaluation et le test des produits pour ce qui est de l'application à utiliser et de l'exécution de cette application. Ni la société Schneider Electric ni aucune de ses sociétés affiliées ou filiales ne peuvent être tenues pour responsables de la mauvaise utilisation des informations contenues dans le présent document. Si vous avez des suggestions, des améliorations ou des corrections à apporter à cette publication, veuillez nous en informer.

Vous acceptez de ne pas reproduire, excepté pour votre propre usage à titre non commercial, tout ou partie de ce document et sur quelque support que ce soit sans l'accord écrit de Schneider Electric. Vous acceptez également de ne pas créer de liens hypertextes vers ce document ou son contenu. Schneider Electric ne concède aucun droit ni licence pour l'utilisation personnelle et non commerciale du document ou de son contenu, sinon une licence non exclusive pour une consultation « en l'état », à vos propres risques. Tous les autres droits sont réservés.

Toutes les réglementations locales, régionales et nationales pertinentes doivent être respectées lors de l'installation et de l'utilisation de ce produit. Pour des raisons de sécurité et afin de garantir la conformité aux données système documentées, seul le fabricant est habilité à effectuer des réparations sur les composants.

Lorsque des équipements sont utilisés pour des applications présentant des exigences techniques de sécurité, suivez les instructions appropriées.

La non-utilisation du logiciel Schneider Electric ou d'un logiciel approuvé avec nos produits matériels peut entraîner des blessures, des dommages ou un fonctionnement incorrect.

Le non-respect de cette consigne peut entraîner des lésions corporelles ou des dommages matériels.

© 2018 Schneider Electric. Tous droits réservés.

---

# Table des matières

---



	Consignes de sécurité .....	7
	A propos de ce manuel .....	11
<b>Partie I</b>	<b>Mise en œuvre matérielle des communications</b>	
	<b>Modbus Serial et Mode caractère .....</b>	<b>15</b>
<b>Chapitre 1</b>	<b>Présentation des communications série .....</b>	<b>17</b>
	Communication Modbus série et Mode caractère .....	18
	Présentation de la liaison série sur les processeurs Modicon M340 ..	19
	Normes et certifications .....	23
	Considérations relatives au câblage .....	24
<b>Chapitre 2</b>	<b>Architectures de communication série .....</b>	<b>25</b>
	Terminaison de ligne Modbus et polarisation (RS485) .....	26
	Raccordement d'équipements Modbus (RS485) .....	28
	Raccordement d'un équipement terminal de données (DTE) (RS232)	31
	Raccordement d'un équipement terminal de circuit de données (DCE)	
	(RS232) .....	33
	Câblage .....	36
<b>Partie II</b>	<b>Mise en œuvre logicielle des communications</b>	
	<b>Modbus Série et Mode caractère .....</b>	<b>41</b>
<b>Chapitre 3</b>	<b>Méthodologie de mise en œuvre .....</b>	<b>43</b>
	Présentation de la phase d'installation .....	43
<b>Chapitre 4</b>	<b>Communication Modbus série pour les processeurs</b>	
	<b>Modicon M340 .....</b>	<b>45</b>
4.1	Généralités .....	46
	A propos de Modbus Série .....	47
	Performances .....	48
	Comment accéder aux paramètres de la liaison série .....	50
4.2	Configuration d'une communication Modbus Serial .....	53
	Ecran de configuration de la communication Modbus série .....	54
	Paramètres Modbus liés à l'application .....	56
	Paramètres des signaux et de la ligne physique en mode Modbus ..	58
	Paramètres Modbus liés à la transmission .....	60
4.3	Programmation d'une communication Modbus Serial .....	62
	Services pris en charge par le processeur maître d'une liaison Modbus	63
	Services pris en charge par le processeur esclave d'une liaison	
	Modbus .....	65

4.4	Mise au point d'une communication Modbus Serial . . . . .	67
	Ecran de mise au point d'une communication Modbus Série . . . . .	67
<b>Chapitre 5</b>	<b>Communication Mode caractère pour les processeurs</b>	
	<b>Modicon M340 . . . . .</b>	<b>69</b>
5.1	Généralités. . . . .	70
	A propos de la communication Mode caractère. . . . .	71
	Performances. . . . .	72
5.2	Configuration d'une communication en mode caractère . . . . .	73
	Ecran de configuration d'une communication Mode caractère . . . . .	74
	Paramètres de détection de fin de message en mode caractère. . . . .	76
	Paramètres des signaux et de la ligne physique en mode caractère. . . . .	78
	Paramètres de la transmission Mode caractère. . . . .	80
5.3	Programmation d'une communication en mode caractère. . . . .	82
	Fonctions de communication Mode caractère . . . . .	82
5.4	Mise au point d'une communication en mode caractère . . . . .	84
	Ecran de mise au point d'une communication Mode caractère . . . . .	84
<b>Chapitre 6</b>	<b>Objets langage des communications Modbus et Mode</b>	
	<b>caractère . . . . .</b>	<b>87</b>
6.1	Objets langage et IODDT des communications Modbus et Mode	
	caractère . . . . .	88
	Présentation des objets langage pour les communications Modbus et	
	Mode caractère . . . . .	89
	Objets langage à échanges implicites associés à la fonction métier . . . . .	90
	Objets langage à échanges explicites associés à la fonction métier . . . . .	91
	Gestion des échanges et des comptes rendus avec des objets	
	explicites . . . . .	93
6.2	Objets langage et IODDT génériques pour les protocoles de	
	communication. . . . .	96
	Informations détaillées sur les objets à échange implicite IODDT de	
	type T_COM_STS_GEN. . . . .	97
	Détails sur les objets à échange explicite de l'IODDT de type	
	T_COM_STS_GEN . . . . .	98
6.3	Objets langage et IODDT associés aux communications Modbus . . . . .	100
	Informations détaillées sur les objets langage à échange explicite pour	
	une fonction Modbus . . . . .	101
	Informations détaillées sur les objets à échanges explicites des IODDT	
	de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT . . . . .	102
	Informations détaillées sur les objets à échanges explicites IODDT de	
	types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT . . . . .	103
	Informations détaillées sur les objets langage associés au mode de	
	communication Modbus . . . . .	106

6.4	Objets langage et IODDT associés aux communications Modbus en mode caractère . . . . .	108
	Informations détaillées sur les objets langage à échange explicite pour la communication en Mode caractère. . . . .	109
	Informations détaillées sur les objets à échanges implicites IODDT de type T_COM_CHAR_BMX. . . . .	110
	Informations détaillées sur les objets à échanges explicites de l'IODDT de type T_COM_CHAR_BMX . . . . .	111
	Informations détaillées sur les objets langage associés à la configuration en mode caractère . . . . .	114
6.5	Type d'IODDT Type T_GEN_MOD applicable à tous les modules . . . . .	116
	Informations détaillées sur les objets langage de l'IODDT de type T_GEN_MOD. . . . .	116
<b>Chapitre 7</b>	<b>Changement dynamique de protocoles . . . . .</b>	<b>119</b>
	Changement de protocole avec les processeurs Modicon M340 . . . . .	119
<b>Partie III</b>	<b>Mise en route : Exemple de mise en œuvre d'une liaison série . . . . .</b>	<b>123</b>
<b>Chapitre 8</b>	<b>Description de l'application . . . . .</b>	<b>125</b>
	Vue d'ensemble de l'application . . . . .	125
<b>Chapitre 9</b>	<b>Installation de l'application avec Control Expert . . . . .</b>	<b>127</b>
9.1	Présentation de la solution utilisée . . . . .	128
	Différentes étapes du processus utilisant Control Expert . . . . .	128
9.2	Développement de l'application. . . . .	129
	Création du projet . . . . .	130
	Déclaration des variables . . . . .	135
	Utilisation d'un modem . . . . .	139
	Procédure de programmation . . . . .	141
	Structure de programmation . . . . .	143
	Programmation . . . . .	146
<b>Chapitre 10</b>	<b>Démarrage de l'application . . . . .</b>	<b>155</b>
	Exécution de l'application en mode Standard . . . . .	155
<b>Glossaire</b>	. . . . .	<b>159</b>
<b>Index</b>	. . . . .	<b>167</b>



# Consignes de sécurité



## Informations importantes

### AVIS

Lisez attentivement ces instructions et examinez le matériel pour vous familiariser avec l'appareil avant de tenter de l'installer, de le faire fonctionner, de le réparer ou d'assurer sa maintenance. Les messages spéciaux suivants que vous trouverez dans cette documentation ou sur l'appareil ont pour but de vous mettre en garde contre des risques potentiels ou d'attirer votre attention sur des informations qui clarifient ou simplifient une procédure.



La présence de ce symbole sur une étiquette "Danger" ou "Avertissement" signale un risque d'électrocution qui provoquera des blessures physiques en cas de non-respect des consignes de sécurité.



Ce symbole est le symbole d'alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

## DANGER

**DANGER** signale un risque qui, en cas de non-respect des consignes de sécurité, **provoque** la mort ou des blessures graves.

## AVERTISSEMENT

**AVERTISSEMENT** signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** la mort ou des blessures graves.

## ATTENTION

**ATTENTION** signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** des blessures légères ou moyennement graves.

## AVIS

**AVIS** indique des pratiques n'entraînant pas de risques corporels.

---

## REMARQUE IMPORTANTE

L'installation, l'utilisation, la réparation et la maintenance des équipements électriques doivent être assurées par du personnel qualifié uniquement. Schneider Electric décline toute responsabilité quant aux conséquences de l'utilisation de ce matériel.

Une personne qualifiée est une personne disposant de compétences et de connaissances dans le domaine de la construction, du fonctionnement et de l'installation des équipements électriques, et ayant suivi une formation en sécurité leur permettant d'identifier et d'éviter les risques encourus.

## AVANT DE COMMENCER

N'utilisez pas ce produit sur les machines non pourvues de protection efficace du point de fonctionnement. L'absence de ce type de protection sur une machine présente un risque de blessures graves pour l'opérateur.

### AVERTISSEMENT

#### EQUIPEMENT NON PROTEGE

- N'utilisez pas ce logiciel ni les automatismes associés sur des appareils non équipés de protection du point de fonctionnement.
- N'accédez pas aux machines pendant leur fonctionnement.

**Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

Cet automatisme et le logiciel associé permettent de commander des processus industriels divers. Le type ou le modèle d'automatisme approprié pour chaque application dépendra de facteurs tels que la fonction de commande requise, le degré de protection exigé, les méthodes de production, des conditions inhabituelles, la législation, etc. Dans certaines applications, plusieurs processeurs seront nécessaires, notamment lorsque la redondance de sauvegarde est requise.

Vous seul, en tant que constructeur de machine ou intégrateur de système, pouvez connaître toutes les conditions et facteurs présents lors de la configuration, de l'exploitation et de la maintenance de la machine, et êtes donc en mesure de déterminer les équipements automatisés, ainsi que les sécurités et verrouillages associés qui peuvent être utilisés correctement. Lors du choix de l'automatisme et du système de commande, ainsi que du logiciel associé pour une application particulière, vous devez respecter les normes et réglementations locales et nationales en vigueur. Le document National Safety Council's Accident Prevention Manual (reconnu aux Etats-Unis) fournit également de nombreuses informations utiles.

Dans certaines applications, telles que les machines d'emballage, une protection supplémentaire, comme celle du point de fonctionnement, doit être fournie pour l'opérateur. Elle est nécessaire si les mains ou d'autres parties du corps de l'opérateur peuvent entrer dans la zone de point de pincement ou d'autres zones dangereuses, risquant ainsi de provoquer des blessures graves. Les produits logiciels seuls, ne peuvent en aucun cas protéger les opérateurs contre d'éventuelles blessures. C'est pourquoi le logiciel ne doit pas remplacer la protection de point de fonctionnement ou s'y substituer.



---

Avant de mettre l'équipement en service, assurez-vous que les dispositifs de sécurité et de verrouillage mécaniques et/ou électriques appropriés liés à la protection du point de fonctionnement ont été installés et sont opérationnels. Tous les dispositifs de sécurité et de verrouillage liés à la protection du point de fonctionnement doivent être coordonnés avec la programmation des équipements et logiciels d'automatisation associés.

**NOTE :** La coordination des dispositifs de sécurité et de verrouillage mécaniques/électriques du point de fonctionnement n'entre pas dans le cadre de cette bibliothèque de blocs fonction, du Guide utilisateur système ou de toute autre mise en œuvre référencée dans la documentation.

## DEMARRAGE ET TEST

Avant toute utilisation de l'équipement de commande électrique et des automatismes en vue d'un fonctionnement normal après installation, un technicien qualifié doit procéder à un test de démarrage afin de vérifier que l'équipement fonctionne correctement. Il est essentiel de planifier une telle vérification et d'accorder suffisamment de temps pour la réalisation de ce test dans sa totalité.

### AVERTISSEMENT

#### RISQUES INHERENTS AU FONCTIONNEMENT DE L'EQUIPEMENT

- Assurez-vous que toutes les procédures d'installation et de configuration ont été respectées.
- Avant de réaliser les tests de fonctionnement, retirez tous les blocs ou autres cales temporaires utilisés pour le transport de tous les dispositifs composant le système.
- Enlevez les outils, les instruments de mesure et les débris éventuels présents sur l'équipement.

**Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

Effectuez tous les tests de démarrage recommandés dans la documentation de l'équipement. Conservez toute la documentation de l'équipement pour référence ultérieure.

#### **Les tests logiciels doivent être réalisés à la fois en environnement simulé et réel.**

Vérifiez que le système entier est exempt de tout court-circuit et mise à la terre temporaire non installée conformément aux réglementations locales (conformément au National Electrical Code des Etats-Unis, par exemple). Si des tests diélectriques sont nécessaires, suivez les recommandations figurant dans la documentation de l'équipement afin d'éviter de l'endommager accidentellement.

Avant de mettre l'équipement sous tension :

- Enlevez les outils, les instruments de mesure et les débris éventuels présents sur l'équipement.
- Fermez le capot du boîtier de l'équipement.
- Retirez toutes les mises à la terre temporaires des câbles d'alimentation entrants.
- Effectuez tous les tests de démarrage recommandés par le fabricant.

---

## FONCTIONNEMENT ET REGLAGES

Les précautions suivantes sont extraites du document NEMA Standards Publication ICS 7.1-1995 (la version anglaise prévaut) :

- Malgré le soin apporté à la conception et à la fabrication de l'équipement ou au choix et à l'évaluation des composants, des risques subsistent en cas d'utilisation inappropriée de l'équipement.
- Il arrive parfois que l'équipement soit dérégulé accidentellement, entraînant ainsi un fonctionnement non satisfaisant ou non sécurisé. Respectez toujours les instructions du fabricant pour effectuer les réglages fonctionnels. Les personnes ayant accès à ces réglages doivent connaître les instructions du fabricant de l'équipement et les machines utilisées avec l'équipement électrique.
- Seuls ces réglages fonctionnels, requis par l'opérateur, doivent lui être accessibles. L'accès aux autres commandes doit être limité afin d'empêcher les changements non autorisés des caractéristiques de fonctionnement.

---

# A propos de ce manuel

---



## Présentation

### Objectif du document

Ce manuel décrit le principe de mise en œuvre matérielle et logicielle du mode caractère et de la communication Modbus pour les automates Modicon M340.

### Champ d'application

Cette documentation est applicable à EcoStruxure™ Control Expert 14.0 ou version ultérieure.

Les caractéristiques techniques des équipements décrits dans ce document sont également fournies en ligne. Pour accéder à ces informations en ligne :

Etape	Action
1	Accédez à la page d'accueil de Schneider Electric <a href="http://www.schneider-electric.com">www.schneider-electric.com</a> .
2	Dans la zone <b>Search</b> , saisissez la référence d'un produit ou le nom d'une gamme de produits. <ul style="list-style-type: none"><li>● N'insérez pas d'espaces dans la référence ou la gamme de produits.</li><li>● Pour obtenir des informations sur un ensemble de modules similaires, utilisez des astérisques (*).</li></ul>
3	Si vous avez saisi une référence, accédez aux résultats de recherche <b>Product Datasheets</b> et cliquez sur la référence qui vous intéresse. Si vous avez saisi une gamme de produits, accédez aux résultats de recherche <b>Product Ranges</b> et cliquez sur la gamme de produits qui vous intéresse.
4	Si plusieurs références s'affichent dans les résultats de recherche <b>Products</b> , cliquez sur la référence qui vous intéresse.
5	Selon la taille de l'écran, vous serez peut-être amené à faire défiler la page pour consulter la fiche technique.
6	Pour enregistrer ou imprimer une fiche technique au format .pdf, cliquez sur <b>Download XXX product datasheet</b> .

Les caractéristiques présentées dans ce document devraient être identiques à celles fournies en ligne. Toutefois, en application de notre politique d'amélioration continue, nous pouvons être amenés à réviser le contenu du document afin de le rendre plus clair et plus précis. Si vous constatez une différence entre le document et les informations fournies en ligne, utilisez ces dernières en priorité.

## Documents à consulter

Titre du document	Numéro de référence
Modicon X80 - Module de liaison série BMXNOM0200 - Manuel utilisateur	EIO0000002696 (anglais), EIO0000002697 (français), EIO0000002698 (allemand), EIO0000002699 (italien), EIO0000002700 (espagnol), EIO0000002701 (chinois)
Plateformes, normes et certifications Modicon M580, M340 et X80 I/O	EIO0000002726 (anglais), EIO0000002727 (français), EIO0000002728 (allemand), EIO0000002730 (italien), EIO0000002729 (espagnol), EIO0000002731 (chinois)
EcoStruxure™ Control Expert - Modes de fonctionnement	33003101 (anglais), 33003102 (français), 33003103 (allemand), 33003104 (espagnol), 33003696 (italien), 33003697 (chinois)
EcoStruxure™ Control Expert - Communication - Bibliothèque de blocs	33002527 (anglais), 33002528 (français), 33002529 (allemand), 33003682 (italien), 33002530 (espagnol), 33003683 (chinois)
EcoStruxure™ Control Expert - Gestion des E/S - Bibliothèque de blocs	33002531 (anglais), 33002532 (français), 33002533 (allemand), 33003684 (italien), 33002534 (espagnol), 33003685 (chinois)

Vous pouvez télécharger ces publications ainsi que d'autres informations techniques sur notre site Web : [www.schneider-electric.com/en/download](http://www.schneider-electric.com/en/download).

---

## Information spécifique au produit

### AVERTISSEMENT

#### FONCTIONNEMENT IMPREU DE L'EQUIPEMENT

L'utilisation de ce produit requiert une expertise dans la conception et la programmation des systèmes d'automatisme. Seules les personnes avec l'expertise adéquate sont autorisées à programmer, installer, modifier et utiliser ce produit.

Respectez toutes les réglementations et normes de sécurité locales et nationales.

**Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**



---

# Partie I

## Mise en œuvre matérielle des communications Modbus Serial et Mode caractère

---

### Dans cette partie

Cette partie présente la mise en œuvre matérielle des communications Modbus Serial et Mode caractère.

### Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
1	Présentation des communications série	17
2	Architectures de communication série	25





---

# Chapitre 1

## Présentation des communications série

---

### Objet du chapitre

Ce chapitre présente les communications série sur la plate-forme Modicon M340.

Le tableau ci-dessous présente brièvement les deux possibilités permettant la mise en œuvre de communications de liaison série :

Utilisation du port intégré de l'UC M340	Utilisation du module de communication BMX NOM 0200 ( <i>voir Modicon X80, Module de liaison série BMXNOM0200, Manuel de l'utilisateur</i> )
Vitesse de transmission limitée Liaisons série non isolées Alimentation de l'équipement terminal	Nombre accru de voies de communication disponibles Traitement des signaux RS232 propres au modem Vitesse de transmission supérieure Deux liaisons série RS485 isolées

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Communication Modbus série et Mode caractère	18
Présentation de la liaison série sur les processeurs Modicon M340	19
Normes et certifications	23
Considérations relatives au câblage	24

## Communication Modbus série et Mode caractère

### Général

Les liaisons série prennent en charge deux protocoles de communication :

- Modbus série
- Mode caractère

### Protocole Modbus

Modbus est un protocole standard aux propriétés suivantes :

- Il établit une communication client/serveur entre différents modules d'un bus ou d'une liaison série. Le client est le maître et les modules esclaves sont les serveurs.
- Il repose sur un échange de données à base de requêtes et de réponses offrant des services via différents codes de fonction.
- Il permet d'échanger des trames émises par des applications de type Modbus, dans deux modes :
  - Mode RTU
  - Mode ASCII

La procédure de gestion des échanges est la suivante :

- Un seul équipement peut envoyer des données sur le bus.
- Les échanges sont gérés par le maître. Il est le seul à pouvoir initier des échanges. Les esclaves ne peuvent pas envoyer de messages de leur propre initiative.
- En cas d'échange non valide, le maître répète la requête. L'esclave qui reçoit la requête est déclaré absent par le maître s'il ne répond pas dans un laps de temps donné.
- Si l'esclave ne comprend pas la requête ou ne peut pas la traiter, il renvoie une réponse d'exception au maître. Dans ce cas, le maître peut (ou pas) répéter la requête.

Deux types de dialogue sont possibles entre le maître et un ou plusieurs esclaves :

- Le maître envoie une requête à un numéro d'esclave donné et attend sa réponse.
- Le maître envoie une requête à l'ensemble des esclaves sans attendre une réponse (selon le principe de diffusion générale).

### Communication Mode caractère

Le mode caractère est un mode d'échanges de données point à point entre deux entités. Contrairement au protocole Modbus, il n'établit pas de communications de liaison série hiérarchiquement structurées et ne propose pas de services à l'aide de codes de fonction.

Le mode caractère est asynchrone. Chaque élément d'information textuelle est envoyé ou reçu caractère par caractère à des intervalles de temps irréguliers. Les propriétés suivantes déterminent la durée des échanges :

- un ou deux caractères de fin de trame ;
- le délai d'expiration ;
- le nombre de caractères.

## Présentation de la liaison série sur les processeurs Modicon M340

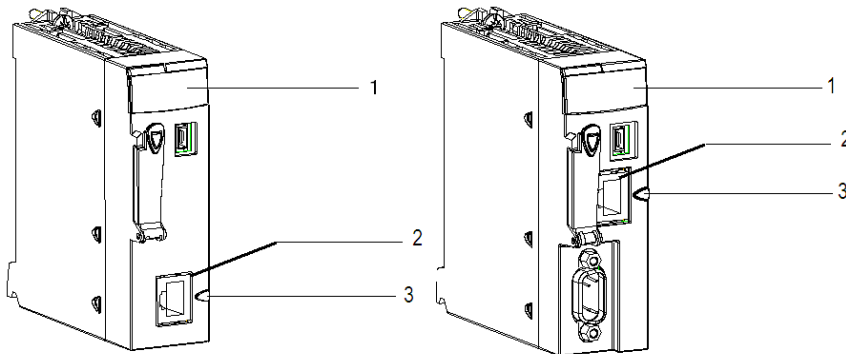
### Général

Les processeurs suivants disposent d'une voie de communication intégrée, dédiée aux communications série, et peuvent communiquer via une liaison série :

- BMX P34 1000
- BMX P34 2000
- BMX P34 2010
- BMX P34 20102
- BMX P34 2020

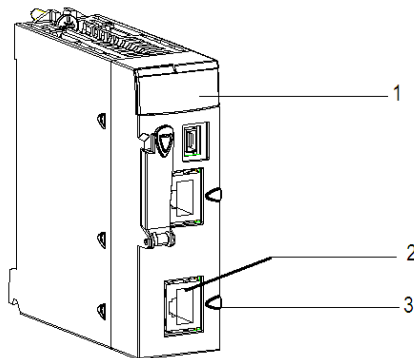
### Emplacement du port série

Les figures ci-dessous montrent l'emplacement du port série sur les processeurs Modicon M340 :



BMX P34 1000/2000

BMX P34 2010/20102



BMX P34 2020

Ces processeurs comprennent les éléments suivants :

Adresse	Description
1	Voyants d'état du processeur à l'avant
2	Voie intégrée (voie 0) dédiée à la liaison série
3	Bague de repérage du port série (noire)

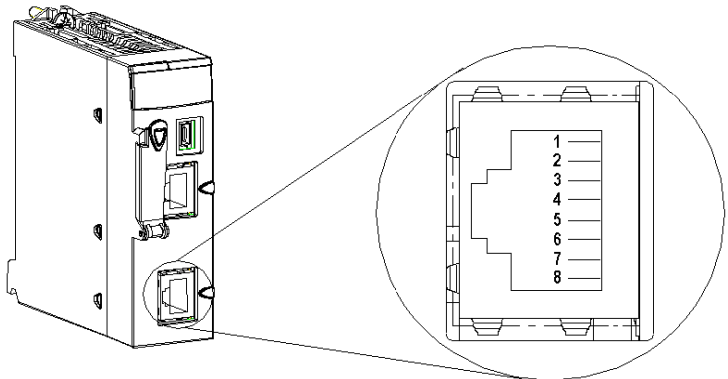
### Diagnostic visuel de la communication série

Le voyant SER COM jaune à l'avant de ces processeurs indique l'état de la communication série :

- Voyant clignotant : communication série en cours.
- Voyant éteint : aucune communication série en cours.

### Description du connecteur du port série

L'illustration suivante représente le port série RJ45 :



Le connecteur RJ45 comporte 8 broches. Les broches utilisées varient selon la liaison physique utilisée.

Les broches utilisées par la liaison série RS 232 sont les suivantes :

- Broche 1 : signal RXD
- Broche 2 : signal TXD
- Broche 3 : signal RTS
- Broche 6 : signal CTS
- Broche 8 : mise à la terre potentielle de la liaison série (0 V)

Les broches utilisées par la liaison série RS 485 sont les suivantes :

- Broche 4 : signal D1
- Broche 5 : signal D0

La broche 7 ne sert qu'à alimenter les interfaces homme-machine ou les petits équipements via le câble de liaison série :

- Broche 7 : alimentation de la liaison série en 5 V CC/190 mA

#### Caractéristiques détaillées

##### Caractéristiques CC :

- Consommation maximum de courant stabilisée : 190 mA
- Tension minimum sur le connecteur de l'UC pour 190 mA : 4,9 V
- Tension maximum sur le connecteur de l'UC pour 190 mA : 5,25 V
- Tension maximum sur le connecteur de l'UC sans charge : 5,5 V.

##### Caractéristiques CA :

- Charge de condensateur : (sur 5 V)
  - 1  $\mu$ F maximum (condensateur en céramique)
  - 10  $\mu$ F (tantale)
- Démarrage du chargement de la pompe : (sur 5 V)
  - 4 x 1  $\mu$ F (condensateur en céramique)
  - 2 x 10  $\mu$ F (tantale)

**NOTE** : les liaisons RS232 à quatre fils, RS485 à deux fils et RS485 à deux fils avec alimentation utilisent le même connecteur RJ45 femelle. Seul le câblage du signal diffère.

### Caractéristiques des lignes électriques

Les liaisons RS232 et RS485 ne sont pas isolées.

En l'absence de terre équipotentielle entre les équipements connectés (longueur de câble supérieure ou égale à 30 m), il est nécessaire d'utiliser un module isolateur TWDXCALISO en mode RS485.

La polarisation de la ligne RS485 est intégrée à l'automate. Le système l'active ou la désactive automatiquement, selon la configuration choisie dans l'écran Control Expert :

- Maître Modbus : la polarisation de la ligne est activée.
- Esclave Modbus : la polarisation de la ligne est désactivée.
- Mode caractère : la polarisation de la ligne est désactivée.

La commutation dynamique de protocole est sans effet sur la polarisation. La valeur des résistances de polarisation est de 560 ohms.

En mode RS232, aucune polarisation n'est nécessaire.

Il n'y a aucune terminaison de ligne intégrée.

### Caractéristiques de la voie

La voie de ces processeurs comprend les éléments suivants :

- une interface physique RS485 non isolée,
- une interface physique RS232 non isolée,
- les types de communication Modbus série (ASCII et RTU) et Mode caractère.

Les caractéristiques de liaison des deux protocoles sont les suivantes :

	<b>Modbus série/RS485</b>	<b>Modbus série/RS232</b>	<b>Mode caractère/RS485</b>	<b>Mode caractère/RS232</b>
Type	Maître/esclave	Maître/esclave	Half Duplex	Full Duplex
Débit	19 200 bauds. Configurable de 300 à 38 400 bauds.	19 200 bauds. Configurable de 300 à 38 400 bauds.	9 600 bauds. Configurable de 300 à 38 400 bauds.	9 600 bauds. Configurable de 300 à 38 400 bauds.
Nombre d'équipements	32	32	–	–
Adresses d'esclave autorisées	1 à 247	1 à 247	–	–
Longueur max. du bus sans branchement	1 000 m (15 m avec branchement)	15 m	1 000 m (15 m avec branchement)	15 m
Taille des messages	Modbus série : <ul style="list-style-type: none"> <li>• RTU : 256 octets (252 octets de données)</li> <li>• ASCII : 513 octets (2 x 252 octets de données)</li> </ul>	Modbus série : <ul style="list-style-type: none"> <li>• RTU : 256 octets (252 octets de données)</li> <li>• ASCII : 513 octets (2 x 252 octets de données)</li> </ul>	1 024 octets	1 024 octets
Services	Lecture de mots/bits. Ecriture de mots/bits. Diagnostics.	Lecture de mots/bits. Ecriture de mots/bits. Diagnostics.	Emission de chaînes de caractères. Réception de chaînes de caractères.	Emission de chaînes de caractères. Réception de chaînes de caractères.

## Normes et certifications

### Aide en ligne

L'aide en ligne de Control Expert vous permet d'accéder aux normes et aux certifications qui s'appliquent aux modules de cette gamme de produits via le guide *Plateformes, normes et certifications Modicon M580, M340 et X80 I/O*.

### Télécharger

Cliquez sur le lien correspondant à votre langue favorite pour télécharger les normes et les certifications (format PDF) qui s'appliquent aux modules de cette gamme de produits :

Langage	
Français	<a href="#"><i>Normes et certifications relatives à Modicon M580, M340 et X80 Aide en ligne</i></a>
Français	<a href="#"><i>Normes et certifications relatives à Modicon M580, M340 et X80 Aide en ligne</i></a>
Allemand	<a href="#"><i>Normes et certifications relatives à Modicon M580, M340 et X80 Aide en ligne</i></a>
Italien	<a href="#"><i>Normes et certifications relatives à Modicon M580, M340 et X80 Aide en ligne</i></a>
Espagnol	<a href="#"><i>Normes et certifications relatives à Modicon M580, M340 et X80 Aide en ligne</i></a>
Chinois	<a href="#"><i>Normes et certifications relatives à Modicon M580, M340 et X80 Aide en ligne</i></a>

## Considérations relatives au câblage

### Considérations opérationnelles

#### **AVERTISSEMENT**

##### **FONCTIONNEMENT INATTENDU DE L'EQUIPEMENT**

Même si vous pouvez connecter ou déconnecter le câble de liaison série sur les UC BMX P34 20x0 pendant que la station est sous tension, cette opération peut interrompre l'application en cours.

**Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

### Liaison

Les situations suivantes peuvent causer une interruption momentanée de l'application ou des communications :

- Le connecteur RJ45 est connecté ou déconnecté sous tension.
- Les modules sont réinitialisés lorsqu'ils sont remis sous tension.



---

# Chapitre 2

## Architectures de communication série

---

### Objet du chapitre

Ce chapitre présente les architectures qui utilisent la communication série, ainsi que les exigences en matière de câblage.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Terminaison de ligne Modbus et polarisation (RS485)	26
Raccordement d'équipements Modbus (RS485)	28
Raccordement d'un équipement terminal de données (DTE) (RS232)	31
Raccordement d'un équipement terminal de circuit de données (DCE) (RS232)	33
Câblage	36

## Terminaison de ligne Modbus et polarisation (RS485)

### Présentation

Un réseau Modbus multipoint doit disposer d'une terminaison de ligne et d'une polarisation.

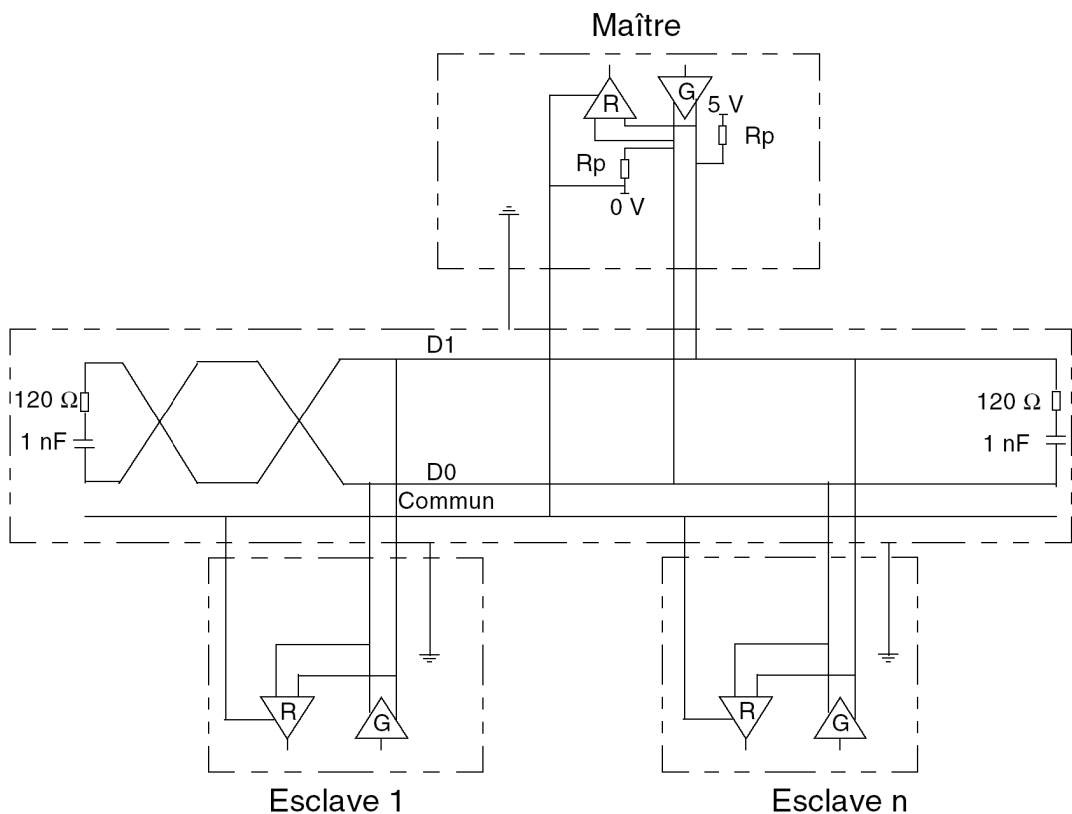
Les équipements susceptibles d'être connectés à ce bus sont :

- autres automates tels que M340, Premium, Quantum, Twido ou Nano,
- appareils Schneider Automation tels que Altivar, module de sécurité XPS, SEPAM, XBT ou Momentum,
- autres appareils compatibles avec le protocole Modbus,
- modem, concentrateur.

Un exemple de **réseau Modbus multipoint** (voir page 29) incluant un processeur BMX P34 2010 est présenté dans ce manuel.

**NOTE** : il est aussi possible de créer un réseau Modbus point à point.

Schéma électrique de terminaison de ligne et de polarisation :



### Terminaison de ligne

La terminaison de ligne est effectuée en externe : elle est composée de deux résistances de  $120 \Omega$  et d'un condensateur de  $1 \text{ nF}$  placés sur chacune des extrémités du réseau (VW3 A8 306 RC ou VW3 A8 306 DRC).

Ne placez pas la terminaison de ligne à l'extrémité d'un câble de dérivation.

### Polarisation de ligne

Sur une ligne Modbus, la polarisation est nécessaire pour un réseau RS485.

- Si l'UC M340 est utilisée comme maître, **elle est automatiquement pilotée par le système** (*voir page 21*) donc il n'y a pas besoin de polarisation externe.
- Si l'UC M340 est utilisée comme esclave, la polarisation doit être mise en œuvre par deux résistances de  $450$  à  $650 \Omega$  ( $R_p$ ) connectées sur la paire équilibrée RS485 :
  - une résistance de démarrage à une tension de  $5 \text{ V}$  sur le circuit D1,
  - une résistance d'arrêt sur le circuit commun D0.

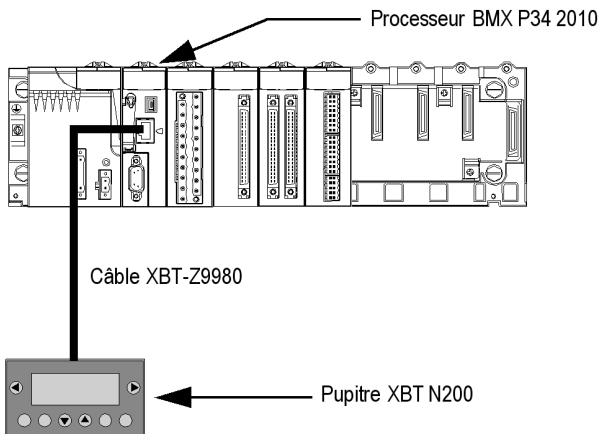
## Raccordement d'équipements Modbus (RS485)

### Général

Les pages suivantes présentent deux exemples de raccordement d'équipements Modbus et une architecture de liaison série Modbus.

### Raccordement d'équipements Modbus alimentés par la liaison série

La figure ci-dessous montre comment un processeur BMX P34 2010 est raccordé à une console XBT N200 alimentée par la liaison série Modbus :



Les équipements sont configurés comme suit :

- Le processeur BMX P34 2010 est configuré comme esclave.
- L'interface homme-machine XBT N200 est configurée comme maître.

Le câble XBT-Z9980 a les propriétés suivantes :

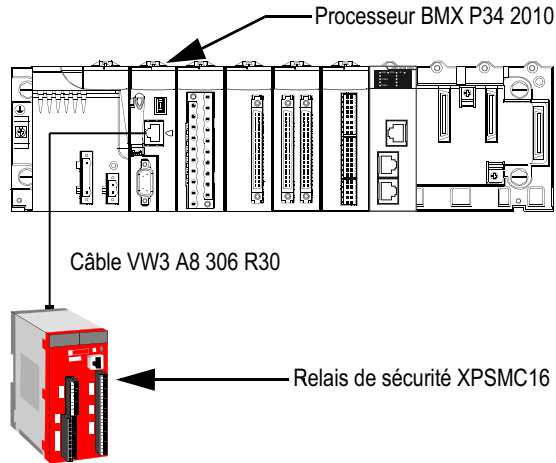
- Raccordement : 2 connecteurs RJ45 mâles
- Câblage : 2 fils pour la ligne physique RS485 et 2 fils pour l'alimentation de la liaison série

### Raccordement d'équipements Modbus non alimentés par la liaison série

Cette architecture comprend les éléments suivants :

- un processeur BMX P34 2010,
- un contrôleur de sécurité XPSMC16.

La figure ci-dessous montre le raccordement d'un processeur BMX P34 2010 à un contrôleur de sécurité XPSMC16 :



Les équipements sont configurés comme suit :

- Le processeur BMX P34 2010 est configuré comme maître.
- Le contrôleur de sécurité XPSMC16 est configuré comme esclave.

Le câble VW3 A8 306 R30 a les propriétés suivantes :

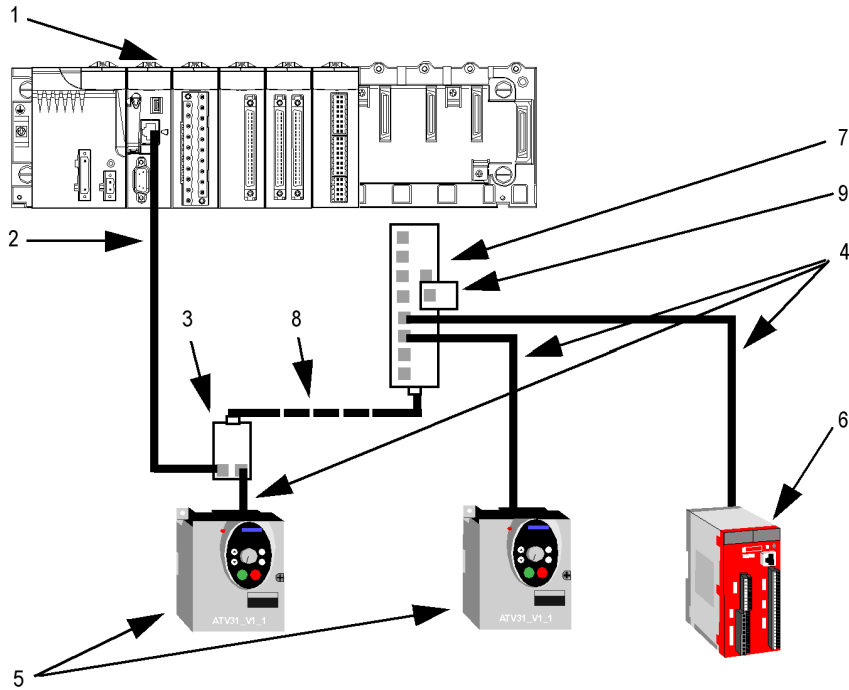
- Raccordement : 2 connecteurs RJ45 mâles
- Câblage : 2 fils pour la ligne physique RS485

### Architecture de liaison Modbus série

L'architecture de liaison Modbus série comprend les éléments suivants :

- un processeur BMX P34 2010/20102 configuré comme maître,
- un contrôleur de sécurité XPSMC16 configuré comme esclave,
- un bloc répartiteur isolé TWDXCAISO,
- un bloc répartiteur LU9 GC3,
- deux variateurs ATV31 configurés comme esclaves.

Le schéma ci-dessous représente l'architecture de liaison série décrite ci-dessus :



- 1 Processeur BMX P34 2010
- 2 Câble XBT-Z9980
- 3 Bloc répartiteur isolé TWDXCAISO
- 4 Câble VW3 A8 306 R30
- 5 Variateur ATV31
- 6 Contrôleur de sécurité XPSMC16
- 7 Bloc répartiteur LU9 GC3
- 8 Câble TSXCSAx00
- 9 Termination de ligne Modbus RC VW3 A8 306

## Raccordement d'un équipement terminal de données (DTE) (RS232)

### Général

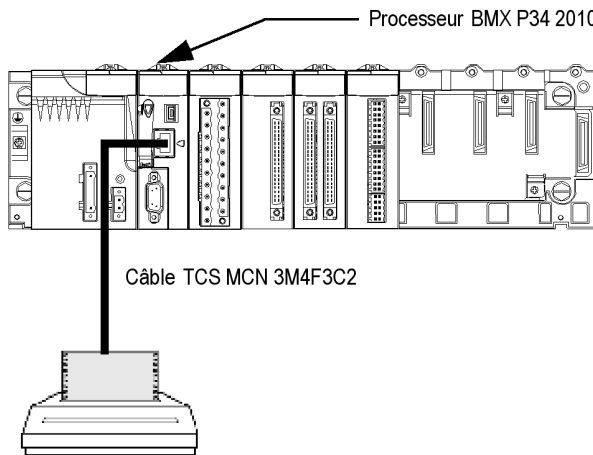
Les équipements terminaux de données sont des équipements tels que :

- périphériques usuels (imprimante, écran-clavier, terminal d'atelier...),
- périphériques spécialisés (lecteur de codes barres par exemple),
- ordinateur PC.

Tous les équipements terminaux de données sont raccordés à un processeur BMX P34 1000/2000/2010/20102/2020 par un câble série croisé utilisant la ligne physique RS232.

### Raccordement d'un équipement terminal de données

La figure suivante représente le raccordement d'une imprimante à un processeur BMX P34 2010 :



La communication utilisée est une communication en Mode caractère.

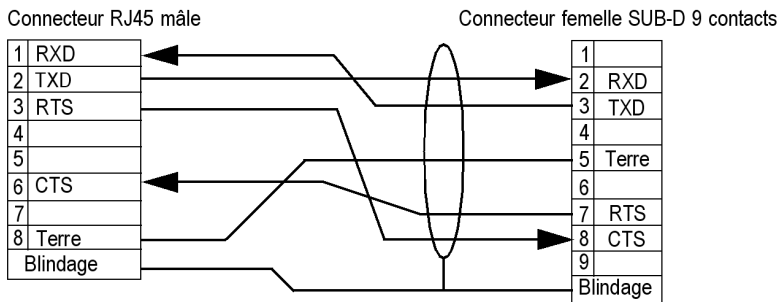
**NOTE :** Un seul équipement terminal de données peut être raccordé à un processeur BMX P34 1000/2000/2010/20102/2020.

### Câble série croisé RS 232

Le câble série croisé TCS MCN 3M4F3C2 comporte 2 connecteurs :

- RJ45 mâle,
- SUB-D 9 contacts femelle.

La figure suivante représente le brochage d'un câble série croisé TCS MCN 3M4F3C2 :



### Câbles et accessoires de raccordement

Le tableau suivant présente les références des câbles et des adaptateurs à utiliser en fonction du connecteur série de l'équipement terminal de données :

Connecteur série de l'équipement terminal de données	Câblage
Connecteur mâle SUB-D 9 contacts	Câble TCS MCN 3M4F3C2
Connecteur SUB-D 25 contacts mâle	<ul style="list-style-type: none"> <li>● Câble TCS MCN 3M4F3C2</li> <li>● Adaptateur TSX CTC 07</li> </ul>
Connecteur femelle SUB-D 25 contacts	<ul style="list-style-type: none"> <li>● Câble TCS MCN 3M4F3C2</li> <li>● Adaptateur TSX CTC 10</li> </ul>



## Raccordement d'un équipement terminal de circuit de données (DCE) (RS232)

### Général

Les équipements terminaux de circuit de données représentent des équipements tels que des modems.

Pour un appareil de type DCE, les broches RTS et CTS sont reliées directement (et non pas croisées).

Tous les équipements terminaux de données sont raccordés à un processeur BMX P34 1000/2000/2010/20102/2020 par un câble série direct utilisant la ligne physique RS232.

**NOTE :** Les différences entre les branchements DCE et DTE sont essentiellement dans les fiches et la direction du signal des broches (entrée ou sortie). Par exemple, un PC de bureau est appelé périphérique DTE alors qu'un modem est appelé périphérique DCE.

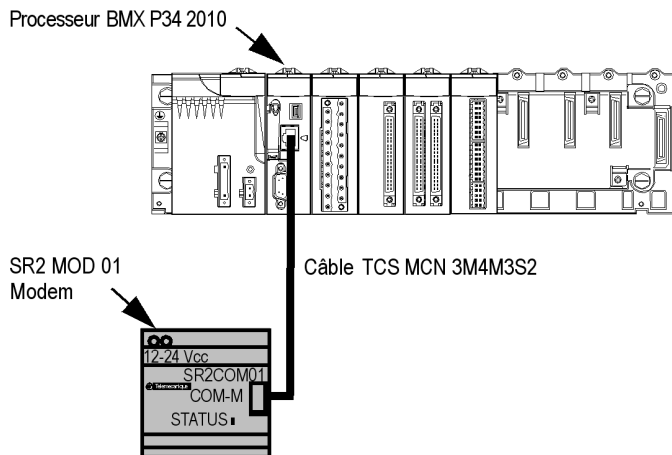
### Caractéristiques d'un modem

Les UC M340 travaillent avec la plupart des modems du marché. Pour raccorder un modem au port série d'un processeur BMX P34 1000/2000/2010/20102/2020, ce modem doit impérativement avoir les caractéristiques suivantes :

- prendre en charge le format 10 bits ou 11 bits par caractère si le port terminal est utilisé pour Modbus Serial :
  - 7 ou 8 bits de donnée,
  - 1 ou 2 bits d'arrêt,
  - parité impaire, paire ou sans parité.
- fonctionner sans contrôle de porteuse.

### Raccordement d'un équipement terminal de circuit de données

La figure suivante représente le raccordement d'un modem à un processeur BMX P34 2010 :



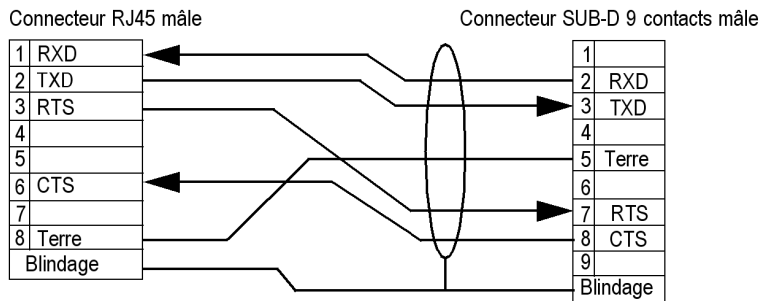
**NOTE :** En Modbus Serial, le délai d'attente doit être compris entre 100 et 250 ms.

### Câble série direct RS 232

Le câble série direct TCS MCN 3M4M3S2 comporte 2 connecteurs :

- RJ45 mâle,
- SUB-D 9 broches mâle.

La figure suivante représente le brochage d'un câble série direct TCS MCN 3M4M3S2 :



### Câbles et accessoires de raccordement

Le tableau suivant présente les références des câbles et des adaptateurs à utiliser en fonction du connecteur série de l'équipement terminal de circuit de données :

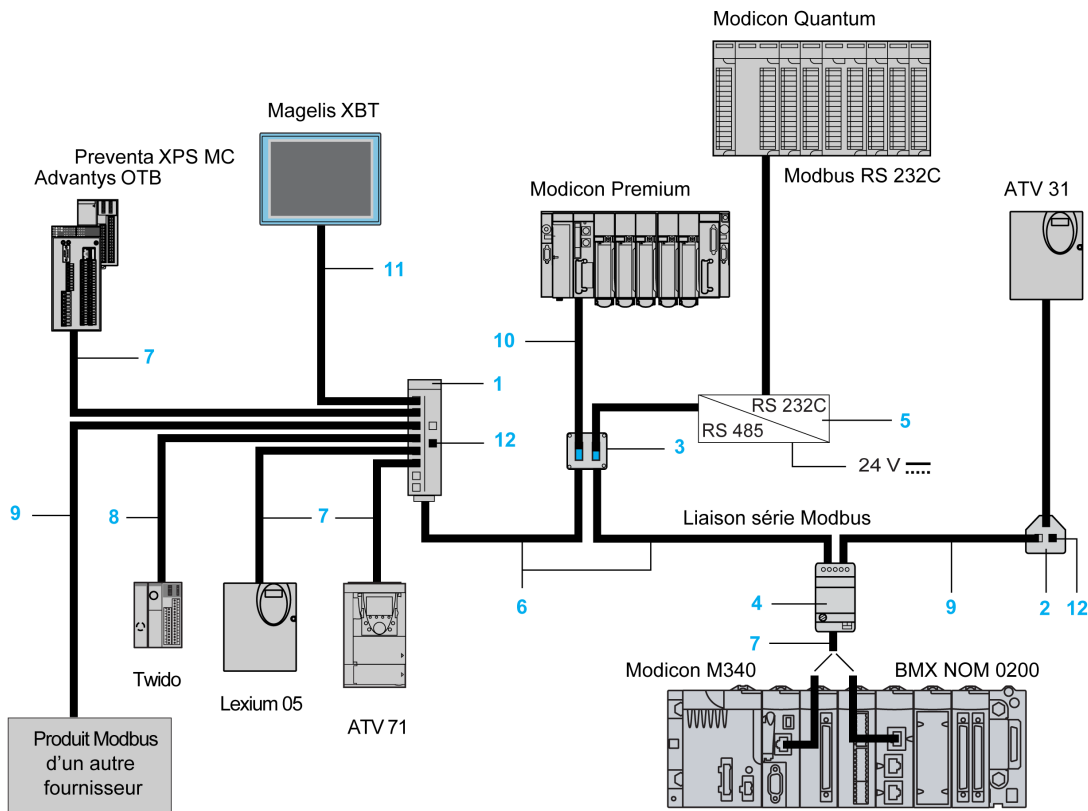
<b>Connecteur série de l'équipement connecté de données</b>	<b>Câblage</b>
Connecteur femelle SUB-D 9 broches	Câble TCS MCN 3M4M3S2
Connecteur femelle SUB-D 25 broches	<ul style="list-style-type: none"><li>● Câble TCS MCN 3M4M3S2</li><li>● Adaptateur TSX CTC 09</li></ul>

## Câblage

### Système de câblage

Plusieurs câbles et accessoires sont nécessaires pour configurer une liaison série.

La figure ci-dessous montre un exemple de liaison Modbus série et de système de câblage en Mode caractère. Les **câbles** (voir page 37) et **accessoires de raccordement** (voir page 38) référencés dans la figure sont décrits dans les tableaux suivants :



## Câbles

Le tableau ci-dessous montre les câbles disponibles qui sont compatibles avec la communication série sur ces processeurs et ce module :

Référence sur la figure	Désignation	Caractéristiques	Longueur	Réf. commerciale
6	Câble principal à paires torsadées à double blindage RS485	2 extrémités dénudées	100 m	TSX CSA 100
			200 m	TSX CSA 200
			500 m	TSX CSA 500
7	Câble RS485 Modbus	2 connecteurs RJ45 mâles	0,3 m	VW3 A8 306 R03
			1 m	VW3 A8 306 R10
			3 m	VW3 A8 306 R30
-	Câble RS485 Modbus	<ul style="list-style-type: none"> <li>● 1 connecteur RJ45 mâle</li> <li>● 1 connecteur SUB-D mâle à 15 broches</li> </ul>	3 m	VW3 A8 306
8	Câble RS485 Modbus	<ul style="list-style-type: none"> <li>● 1 connecteur RJ45 mâle</li> <li>● 1 connecteur mini-DIN</li> </ul>	0,3 m	TWD XCA RJ003
			1 m	TWD XCA RJ010
			3 m	TWD XCA RJ030
9	Câble RS485 Modbus	<ul style="list-style-type: none"> <li>● 1 connecteur RJ45 mâle</li> <li>● 1 extrémité dénudée</li> </ul>	3 m	VW3 A8 306 D30
10	Câble RS485 Modbus	<ul style="list-style-type: none"> <li>● 1 connecteur miniature</li> <li>● 1 connecteur SUB-D à 15 broches</li> </ul>	3 m	TSX SCP CM 4630
11	Câble RS485 pour afficheur et terminal Magelis XBT	<ul style="list-style-type: none"> <li>● 1 connecteur RJ45 mâle</li> <li>● 1 connecteur SUB-D femelle à 25 broches</li> </ul> <p><b>Remarque :</b> ce câble n'est pas compatible avec le module BMX NOM 0200.</p>	2,5 m	XBT-Z938
-	Câble RS485 pour équipements alimentés via la liaison série	2 connecteurs RJ45 mâles <p><b>Remarque :</b> ce câble n'est pas compatible avec le module BMX NOM 0200.</p>	3 m	XBT-Z9980
-	Câble RS232 à quatre fils pour équipement DTE (Data Terminal Equipment)	<ul style="list-style-type: none"> <li>● 1 connecteur RJ45 mâle</li> <li>● 1 connecteur SUB-D femelle à 9 broches</li> </ul>	3 m	TCS MCN 3M4F3C2
-	Câble RS232 à quatre fils pour équipement DCE (Data Circuit-terminating Equipment)	<ul style="list-style-type: none"> <li>● 1 connecteur RJ45 mâle</li> <li>● 1 connecteur SUB-D mâle à 9 broches</li> </ul>	3 m	TCS MCN 3M4M3S2

Référence sur la figure	Désignation	Caractéristiques	Longueur	Réf. commerciale
-	Câble RS232 à sept fils pour équipement DCE (Data Circuit-terminating Equipment)	<ul style="list-style-type: none"> <li>● 1 connecteur RJ45 mâle</li> <li>● 1 connecteur SUB-D mâle à 9 broches</li> </ul>	3 m	TCS XCN 3M4F3S4

### Accessoires de raccordement

Le tableau ci-dessous montre les accessoires de raccordement disponibles, qui sont compatibles avec la communication série sur ces processeurs et ce module :

Référence sur la figure	Désignation	Caractéristiques	Réf. commerciale
1	Boîtier répartiteur Modbus	<ul style="list-style-type: none"> <li>● 10 connecteurs RJ45</li> <li>● 1 bornier à vis</li> </ul>	LU9 GC3
2	Boîtier de raccordement en T	<ul style="list-style-type: none"> <li>● 2 connecteurs RJ45</li> <li>● Câble embarqué de 0,3 m avec connecteur RJ45 à l'extrémité</li> </ul>	VW3 A8 306 TF03
		<ul style="list-style-type: none"> <li>● 2 connecteurs RJ45</li> <li>● Câble embarqué de 1 m avec connecteur RJ45 à l'extrémité</li> </ul>	VW3 A8 306 TF10
-	Boîtier de raccordement en T passif	<ul style="list-style-type: none"> <li>● Trois borniers à vis</li> <li>● Adaptateur d'embout RC</li> </ul>	TSX SCA 50
3	Prise d'abonné 2 voies passif	<ul style="list-style-type: none"> <li>● 2 connecteurs SUB-D femelles à 15 broches</li> <li>● 2 borniers à vis</li> <li>● Adaptateur d'embout RC</li> </ul>	TSX SCA 62
4	Boîtier de raccordement en T RS485 isolé	<ul style="list-style-type: none"> <li>● Un connecteur RJ45</li> <li>● 1 bornier à vis</li> </ul>	TWD XCA ISO
-	Boîtier de raccordement en T	3 connecteurs RJ45	TWD XCA T3RJ
-	Adaptateur Modbus/Bluetooth	<ul style="list-style-type: none"> <li>● 1 adaptateur Bluetooth avec 1 connecteur RJ45</li> <li>● 1 cordon pour PowerSuite avec 2 connecteurs RJ45</li> <li>● 1 cordon pour TwidoSuite avec 1 connecteur RJ45 et 1 connecteur mini-DIN</li> <li>● 1 adaptateur RJ45/SUB-D mâle à 9 broches pour variateurs ATV</li> </ul>	VW3 A8 114
5	Adaptateur de ligne RS232C/RS485 sans signaux de modem	19,2 Kbits/s	XGS Z24

Référence sur la figure	Désignation	Caractéristiques	Réf. commerciale
12	Terminaison de ligne pour connecteur RJ45	<ul style="list-style-type: none"> <li>● Résistance de 120 <math>\Omega</math></li> <li>● Capacité de 1 nF</li> </ul>	VW3 A8 306 RC
-	Terminaison de ligne pour bornier à vis	<ul style="list-style-type: none"> <li>● Résistance de 120 <math>\Omega</math></li> <li>● Capacité de 1 nF</li> </ul>	VW3 A8 306 DRC
-	Adaptateur pour équipements non standard	<ul style="list-style-type: none"> <li>● 2 connecteurs SUB-D mâles à 25 broches</li> </ul>	XBT ZG999
-	Adaptateur pour équipements non standard	<ul style="list-style-type: none"> <li>● 1 connecteur SUB-D mâle à 25 broches</li> <li>● 1 connecteur SUB-D mâle à 9 broches</li> </ul>	XBT ZG909
-	Adaptateur pour équipement terminal de données	<ul style="list-style-type: none"> <li>● 1 connecteur SUB-D mâle à 9 broches</li> <li>● 1 connecteur SUB-D femelle à 25 broches</li> </ul>	TSX CTC 07
-	Adaptateur pour équipement terminal de données	<ul style="list-style-type: none"> <li>● 1 connecteur SUB-D mâle à 9 broches</li> <li>● 1 connecteur SUB-D mâle à 25 broches</li> </ul>	TSX CTC 10
-	Adaptateur pour équipement DCE (Data Circuit-terminating Equipment)	<ul style="list-style-type: none"> <li>● 1 connecteur SUB-D femelle à 9 broches</li> <li>● 1 connecteur SUB-D mâle à 25 broches</li> </ul>	TSX CTC 09

**NOTE :** Cette liste de câbles et d'accessoires n'est pas exhaustive.





---

## Partie II

# Mise en œuvre logicielle des communications Modbus Série et Mode caractère

---

### Dans cette partie

Cette partie présente la mise en œuvre logicielle des communications Modbus Série et Mode caractère avec le logiciel Control Expert.

### Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
3	Méthodologie de mise en œuvre	43
4	Communication Modbus série pour les processeurs Modicon M340	45
5	Communication Mode caractère pour les processeurs Modicon M340	69
6	Objets langage des communications Modbus et Mode caractère	87
7	Changement dynamique de protocoles	119



---

# Chapitre 3

## Méthodologie de mise en œuvre

---

### Présentation de la phase d'installation

#### Présentation

L'installation logicielle des modules métiers s'effectue à l'aide des différents éditeurs de Control Expert :

- en mode local
- en mode connecté

Si vous ne disposez pas de processeur auquel vous pouvez vous connecter, Control Expert vous permet d'effectuer un test initial à l'aide du simulateur. Dans ce cas, l'installation est différente.

#### Phases d'installation en cas d'utilisation d'un processeur

Le tableau ci-dessous indique les différentes phases d'installation à l'aide d'un processeur :

Etape	Description	Mode
Configuration du processeur	Déclaration du processeur	Local
	Configuration du port série du processeur	
Configuration du module (le cas échéant)	Déclaration du module	Local
	Configuration de la voie du module	
	Saisie des paramètres de configuration	
Déclaration des variables	Déclaration des variables de type IODDT propres au processeur/module et des variables de projet	Local <sup>(1)</sup>
Association	Association des variables d'IODDT aux voies configurées (éditeur de variables)	Local <sup>(1)</sup>
Programmation	Programmation du projet	Local <sup>(1)</sup>
Génération	Génération du projet (analyse et modification des liens)	Local
Transfert	Transfert du projet vers l'automate	Connecté
Mise au point	Mise au point du projet à l'aide des écrans de mise au point et des tables d'animation	Connecté
Documentation	Création d'un fichier de documentation et impression des différentes informations concernant le projet	Connecté
Fonctionnement	Affichage des différentes informations requises pour superviser le projet	Connecté
<b>(1) Ces phases peuvent également s'effectuer en mode connecté.</b>		

### Phases d'installation en cas d'utilisation d'un simulateur

Le tableau ci-dessous indique les différentes phases d'installation à l'aide d'un simulateur :

Etape	Description	Mode
Configuration du processeur	Déclaration du processeur	Local
	Configuration du port série du processeur	
Configuration du module (le cas échéant)	Déclaration du module	Local
	Configuration de la voie du module	
	Saisie des paramètres de configuration	
Déclaration des variables	Déclaration des variables de type IODDT propres au processeur/module et des variables de projet	Local <sup>(1)</sup>
Association	Association des variables d'IODDT aux voies configurées (éditeur de variables)	Local <sup>(1)</sup>
Programmation	Programmation du projet	Local <sup>(1)</sup>
Génération	Génération du projet (analyse et modification des liens)	Local
Transfert	Transfert du projet vers le simulateur	Connecté
Simulation	Simulation du programme avec des entrées/sorties	Connecté
Réglage/Mise au point	Mise au point du projet à partir de tables d'animation	Connecté
	Modification du programme et des paramètres de réglage	
<b>(1)</b> Ces phases peuvent également s'effectuer en mode connecté.		

### Configuration du processeur et du module

Les paramètres de configuration sont accessibles uniquement depuis le logiciel Control Expert.

### Création de la documentation technique

Control Expert permet de créer une **documentation technique de projet** (voir *EcoStruxure™ Control Expert, Modes de fonctionnement*).

Le format général de l'impression comprend :

- un titre : la référence du module et sa position ;
- une section avec l'identification du module ;
- une section par voie avec tous les paramètres d'une voie.

L'impression est cohérente avec la configuration : les informations non significatives, en gris, ne sont pas imprimées.

---

# Chapitre 4

## Communication Modbus série pour les processeurs Modicon M340

---

### Objet du chapitre

Ce chapitre décrit le processus d'implémentation logicielle de la communication Modbus série pour les processeurs Modicon M340.

### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
4.1	Généralités	46
4.2	Configuration d'une communication Modbus Serial	53
4.3	Programmation d'une communication Modbus Serial	62
4.4	Mise au point d'une communication Modbus Serial	67

# Sous-chapitre 4.1

## Généralités

---

### Objet de cette section

Cette partie présente les généralités sur la communication Modbus Serial et ses services.

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
A propos de Modbus Série	47
Performances	48
Comment accéder aux paramètres de la liaison série	50

## A propos de Modbus Série

### Introduction

La communication par Modbus autorise l'échange de données entre tous les appareils connectés au bus. Modbus Série est un protocole qui crée une structure hiérarchique (un maître et plusieurs esclaves).

Le maître gère l'ensemble des échanges de 2 façons différentes :

- le maître échange avec l'esclave et attend la réponse,
- le maître échange avec l'ensemble des esclaves sans attente de réponse (diffusion générale).

**NOTE** : Veillez à ce que deux maîtres (sur le même bus) n'envoient pas des requêtes simultanément : les demandes seraient perdues et chaque rapport aurait un résultat incorrect qui pourrait être 16#0100 (impossible de traiter la requête) ou 16#ODFF (absence de l'esclave).

### **AVERTISSEMENT**

#### **PERTE DE DONNEES CRITIQUES**

N'utilisez les ports de communication que pour des transferts de données non critiques.

**Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

## Performances

### Présentation

Les tableaux ci-dessous permettent d'évaluer la durée typique des échanges de communication Modbus, selon différents critères.

Les résultats affichés correspondent à la durée moyenne de la fonction `READ_VAR` en millisecondes.

### Définition de la durée d'échange

La durée d'échange est le laps de temps écoulé entre la création d'un échange et la fin de cet échange. Elle inclut le temps de communication sur la liaison série.

L'échange est créé lorsque la fonction de communication est appelée.

Il se termine lorsque l'un des événements ci-dessous survient :

- Des données sont reçues.
- Une anomalie apparaît.
- Un délai arrive à expiration.

### Durée d'échange pour un mot

Le tableau ci-dessous indique les durées d'échange pour un mot de communication Modbus sur un processeur BMX P34 2020 :

Durée d'échange en ms (l'esclave Modbus est un BMX P34 1000 cyclique)		Durée du cycle en ms		
		Cyclique	10	50
Débits en bauds de la communication en bits par seconde	<b>4 800</b>	68	72	100
	<b>9 600</b>	35	40	50
	<b>19 200</b>	20	27	50
	<b>38 400</b>	13	20	50

Les durées d'échange sur un processeur BMX P34 2000/2010/20102 sont similaires à celles sur un processeur BMX P34 2020. Les durées d'échange sur un BMX P34 1000 sont 10 % plus rapides.

**NOTE** : toutes les durées d'échange indiquées ci-dessus proviennent de mesures présentant une marge de précision de +/- 10 ms.



### Durée d'échange pour 100 mots

Le tableau ci-dessous indique les durées d'échange pour 100 mots de communication Modbus sur un processeur BMX P34 2020 :

Durée d'échange en ms (l'esclave Modbus est un BMX P34 1000 cyclique)		Durée du cycle en ms		
		Cyclique	10	50
Débits en bauds de la communication en bits par seconde	<b>4 800</b>	500	540	595
	<b>9 600</b>	280	288	300
	<b>19 200</b>	142	149	150
	<b>38 400</b>	76	80	100

Les durées d'échange sur un processeur BMX P34 2000/2010/20102 sont similaires à celles sur un processeur BMX P34 2020. Les durées d'échange sur un BMX P34 1000 sont 10 % plus rapides.

**NOTE** : toutes les durées d'échange indiquées ci-dessus proviennent de mesures présentant une marge de précision de +/- 10 ms.

## Comment accéder aux paramètres de la liaison série

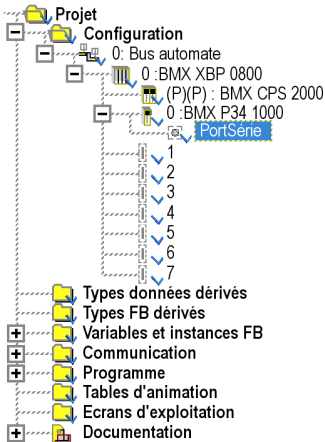
### Présentation

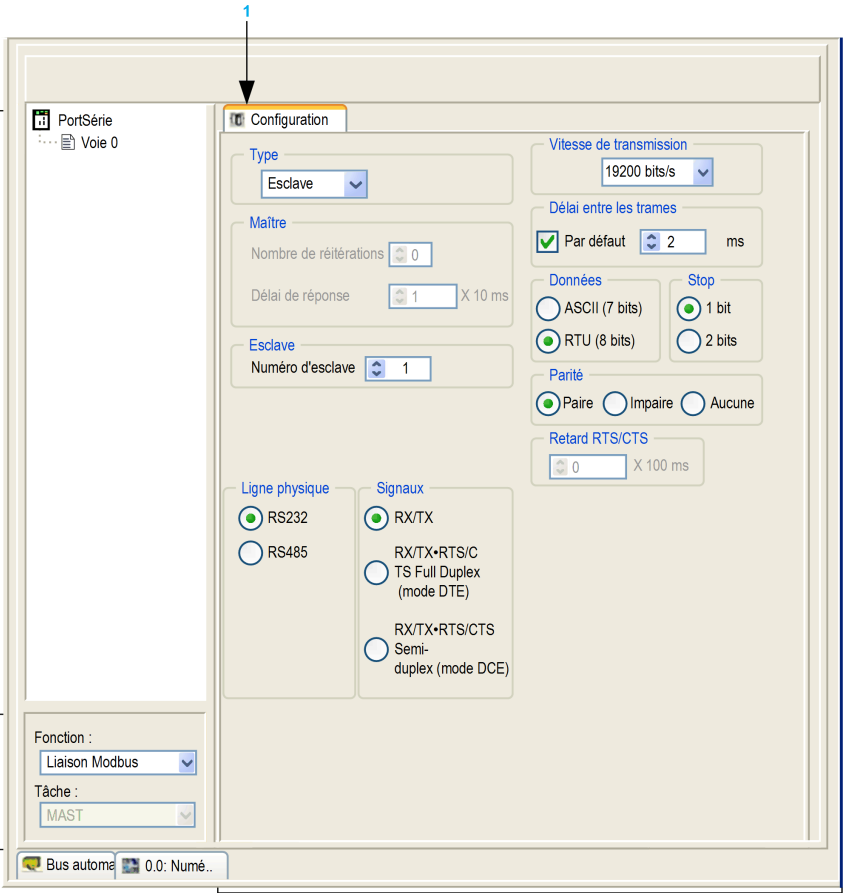
Les pages suivantes expliquent comment accéder à l'écran de configuration du port série pour les processeurs suivants. Vous y trouverez également des informations générales sur les écrans de configuration et de mise au point des liaisons Modbus et Mode caractère :

- BMX P34 1000,
- BMX P34 2000,
- BMX P34 2010/20102,
- BMX P34 2020.

### Comment accéder à la liaison série

Le tableau ci-dessous décrit la marche à suivre pour accéder à la liaison série :

Etape	Action
1	<p>Dans le Navigateur du projet, accédez au répertoire  <i>Projet\Configuration\0: PLC bus\0: BMX XBP ....\0: BMX P34 ....\SerialPort.</i>  <b>Résultat</b> : l'écran suivant s'affiche :</p> 

Etape	Action
2	<p>Double-cliquez sur le répertoire SerialPort.  <b>Résultat</b> : l'écran de configuration s'affiche :</p> 

## Description de l'écran de configuration

Le tableau suivant présente les différents éléments de l'écran de configuration :

Adresse	Élément	Fonction
1	Onglets	L'onglet en avant-plan indique le mode en cours. Chaque mode peut être sélectionné à l'aide de l'onglet correspondant. Les modes suivants sont disponibles : <ul style="list-style-type: none"> <li>● Configuration</li> <li>● Mise au point (accessible seulement en mode connecté)</li> </ul>
2	Zone de voie	Permet : <ul style="list-style-type: none"> <li>● de choisir entre le port série et la voie 0 en cliquant sur l'un ou l'autre ;</li> <li>● d'afficher, en cliquant sur le port série, les onglets suivants : <ul style="list-style-type: none"> <li>○ « Description » qui donne les caractéristiques de l'équipement,</li> <li>○ « Objets d'E/S » (<i>voir EcoStruxure™ Control Expert, Modes de fonctionnement</i>) qui permet de présymboliser les objets d'entrée/sortie ;</li> </ul> </li> <li>● d'afficher, en cliquant sur la voie, les onglets suivants : <ul style="list-style-type: none"> <li>○ Configuration</li> <li>○ Mise au point</li> </ul> </li> <li>● d'afficher le symbole et le nom de la voie définis par l'utilisateur à l'aide de l'éditeur de variables.</li> </ul>
3	Zone des paramètres généraux	Permet de choisir les paramètres généraux associés à la voie : <ul style="list-style-type: none"> <li>● Fonction : les fonctions disponibles sont Liaison Modbus ou Mode caractère. Par défaut, la fonction Modbus est configurée.</li> <li>● Tâche : définit la tâche maître dans laquelle seront échangés les objets à échanges implicites de la voie. Cette zone apparaît en grisé et n'est donc pas configurable.</li> </ul>
4	Zone de configuration ou de mise au point	En mode configuration, cette zone permet de configurer les paramètres de la voie. En mode mise au point, elle permet de mettre au point la voie de communication.

---

## Sous-chapitre 4.2

### Configuration d'une communication Modbus Serial

---

#### Objet de cette section

Cette partie décrit la configuration logicielle d'une communication Modbus Serial.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Ecran de configuration de la communication Modbus série	54
Paramètres Modbus liés à l'application	56
Paramètres des signaux et de la ligne physique en mode Modbus	58
Paramètres Modbus liés à la transmission	60

## Ecran de configuration de la communication Modbus série

### Général

Les pages suivantes présentent l'écran de configuration d'une communication Modbus série.

### Accès à l'écran de configuration

Pour accéder à l'écran de configuration d'une communication Modbus série, ouvrez le répertoire Port série dans le navigateur de projet (*voir page 50*).

### Ecran de configuration d'une communication Modbus série

La figure ci-dessous représente l'écran de configuration d'une communication Modbus série :

The screenshot shows a configuration window titled "Configuration" with the following settings:

- Type:** Esclave
- Vitesse de transmission:** 19200 Bits/s
- Maître:**
  - Nombre de répétitions: 0
  - Délai de réponse: 1 X 10 ms
- Esclave:** Numéro d'esclave: 1
- Délai inter-frames:**  Par défaut 2 ms
- Données:**  ASCII(7 Bits),  RTU(8 Bits)
- Stop:**  1 Bit,  2 Bits
- Parité:**  Paire,  Impaire,  Sans
- Retard RTS/CTS:** 0 X 100 ms
- Ligne physique:**  RS232,  RS485
- Signaux:**  RX/TX,  RX/TX + RTS/CTS DTE mode,  RX/TX + RTS/CTS DCE mode

## Description

Ces zones permettent de configurer les paramètres de voie. En mode connecté, ces zones sont accessibles. En mode local, la zone est accessible mais il se peut que certains paramètres ne soient pas accessibles et apparaissent en grisé.

Le tableau ci-dessous présente les différentes zones de l'écran de configuration de liaison Modbus :

Élément	Commentaire
Paramètres de l'application (voir page 56)	Ces paramètres sont accessibles dans 3 zones : <ul style="list-style-type: none"> <li>● Type</li> <li>● Maître</li> <li>● Esclave</li> </ul>
Paramètres liés aux signaux et à la ligne physique (voir page 58)	Ces paramètres sont accessibles dans 3 zones : <ul style="list-style-type: none"> <li>● Ligne physique</li> <li>● Signaux</li> <li>● Retard RTS/CTS</li> </ul>
Paramètres de la transmission (voir page 60)	Ces paramètres sont accessibles dans 5 zones : <ul style="list-style-type: none"> <li>● Vitesse de transmission</li> <li>● Délai inter-trames</li> <li>● Données</li> <li>● Bits d'arrêt</li> <li>● Parité</li> </ul>

**NOTE** : lors de la configuration de la communication Modbus série en mode maître, la zone **Esclave** est grisée et n'est pas modifiable (et inversement).

## Valeurs par défaut

Le tableau suivant présente les valeurs par défaut des paramètres d'une communication Modbus série :

Paramètres de configuration	Valeur	
Paramètres de l'application	<b>Type</b>	Esclave
	<b>Numéro d'esclave</b>	1
Paramètres des signaux et de la ligne physique	<b>Ligne physique</b>	RS485
	<b>Signaux</b>	RX/TX
Paramètres de la transmission	<b>Vitesse de transmission</b>	19200 bits/s
	<b>Délai entre les trames</b>	2 ms
	<b>Données</b>	RTU (8 bits)
	<b>Arrêter</b>	1 bit
	<b>Parité</b>	Paire

## Paramètres Modbus liés à l'application

### Présentation

Après avoir configuré la voie de communication, vous devez renseigner les paramètres dédiés à l'application.

Ces paramètres sont accessibles à partir de 3 zones :

- la zone Type,
- la zone Maître,
- la zone Esclave.

### Zone Type

Cette zone de configuration apparaît dans l'écran comme représenté ci-dessous :

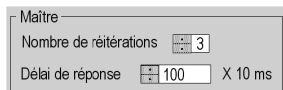


Cette zone vous permet de sélectionner le type de Modbus Serial à utiliser :

- **Maître** : Dans le cas où la station est maître.
- **Esclave** : Dans le cas où la station est esclave.

### Zone Maître

La zone de configuration suivante est accessible uniquement lorsque le type "Maître" est sélectionné :



Cette zone permet de renseigner les paramètres suivants :

- **Nombre de réitérations** : nombre de tentatives de connexion qu'effectue le maître avant de déclarer l'esclave absent.  
La valeur par défaut est 3.  
les valeurs possibles sont comprises entre 0 et 15.  
la valeur 0 indique qu'il n'y a pas de réitération du maître.
- **Délai de réponse** : délai entre la requête émise par le maître et sa réitération en cas de non réponse de l'esclave. Il correspond au temps maximum entre l'émission du dernier caractère de la requête émise par le maître et la réception du premier caractère de la requête renvoyée par l'esclave.  
la valeur par défaut est de 1 seconde (100\*10 ms),  
les valeurs possibles sont comprises entre 10 ms et 10 s.

**NOTE** : le délai de réponse du maître doit être au moins égal au délai de réponse le plus long parmi les esclaves présents sur le bus.



## Zone Esclave

La zone de configuration suivante est accessible uniquement lorsque le type "Esclave" est sélectionné :



The image shows a configuration window titled "Esclave". Inside the window, there is a label "Numéro d'esclave" followed by a text input field containing the number "7".

Cette zone permet de renseigner le numéro d'esclave du processeur :

La valeur par défaut est 1.

les valeurs possibles sont comprises entre 1 et 247.

**NOTE :** dans une configuration Modbus esclave, l'adresse supplémentaire 248 peut être utilisée pour une communication série point à point.

## Paramètres des signaux et de la ligne physique en mode Modbus

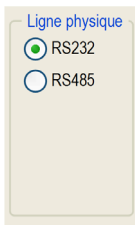
### Présentation

Les paramètres des signaux et de la ligne physique sont accessibles dans trois zones :

- **Ligne physique**
- **Signaux**
- **Retard RTS/CTS**

### Zone Ligne physique

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :

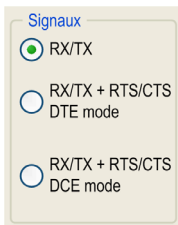


Dans cette zone, vous pouvez choisir entre deux types de ligne physique pour le port série des processeurs BMX P34 1000/2000/2010/20102/2020 :

- la ligne RS232,
- la ligne RS485.

### Zone Signaux

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Dans cette zone, vous pouvez sélectionner les signaux pris en charge par la ligne physique RS232 :

- **RX/TX**
- **RX/TX + RTS/CTS DTE mode**
- **RX/TX + RTS/CTS DCE mode**

Si la ligne physique RS485 est configurée, la zone est entièrement grisée et la valeur par défaut est **RX/TX**.

## Zone Retard RTS/CTS

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



La zone Retard RTS/CTS n'est active que lorsque les cases RS232 et RX/TX+RTS/CTS sont cochées. Un algorithme de contrôle de flux RTS/CTS est sélectionné lorsque la valeur par défaut est 0 ms. Une valeur différente de 0 active un algorithme de contrôle de modem RTS/CTS.

L'algorithme de contrôle de flux RTS/CTS (DTE <-> DTE) est différent de l'algorithme de contrôle de modem RTS/CTS (DTE <-> DCE) :

- L'algorithme de contrôle de flux RTS/CTS est lié au débordement du tampon de réception (duplex intégral).
- L'algorithme de contrôle de modem RTS/CTS est lié au processus d'émission partagée (un modem radio, par exemple).

## Algorithme de contrôle de flux RTS/CTS

L'objectif est d'éviter le dépassement du tampon de réception.

Le signal de sortie RTS de chaque équipement est connecté au signal d'entrée CTS de l'autre équipement. L'émetteur (M340) est autorisé à émettre des données dès qu'il reçoit le signal d'entrée RTS (par exemple, un autre M340) sur son entrée CTS. Cet algorithme est symétrique et permet une communication asynchrone en duplex intégral.

## Algorithme de contrôle de modem RTS/CTS

Avant d'émettre une requête, l'expéditeur (M340) active le signal RTS et attend le signal CTS à déclencher par le modem. Si le signal CTS n'est pas activé dans le délai RTS/CTS imparti, la requête est ignorée.

## Paramètres Modbus liés à la transmission

### Présentation

Les paramètres de transmission sont accessibles dans cinq zones :

- **Vitesse de transmission**
- **Délai entre les trames**
- **Données**
- **Arrêt**
- **Parité**

### Zone Vitesse de transmission

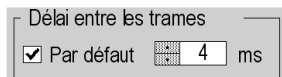
Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Vous pouvez l'utiliser pour sélectionner la vitesse de transmission de la liaison Modbus série. La vitesse sélectionnée doit être cohérente avec les autres équipements. Les valeurs possibles sont 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200 et 38 400 bits par seconde.

### Zone Délai entre les trames

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Le **Délai entre les trames** correspond au temps minimum entre deux trames à la réception. Ce délai est géré lorsque l'automate (maître ou esclave) reçoit des messages.

**NOTE** : la valeur par défaut dépend de la vitesse de transmission sélectionnée.

**NOTE** : pour être conforme à Modbus, le délai entre les trames doit être égal à la valeur Par défaut. En cas de non-conformité d'un esclave, la valeur peut être modifiée et doit être identique pour le maître et l'ensemble des esclaves sur le bus.

## Zone Données

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Cette zone vous permet d'indiquer le type de codage utilisé pour communiquer sur la liaison Modbus série. Ce champ est défini en fonction des autres équipements connectés sur le bus. Les modes disponibles sont au nombre de deux :

- Mode RTU :
  - Les caractères sont codés sur 8 bits.
  - Un silence de 3,5 caractères au moins marque la fin de trame.
  - L'intégrité de la trame est vérifiée à l'aide d'un mot appelé « somme de contrôle CRC », contenu dans la trame.
- Mode ASCII :
  - Les caractères sont codés sur 7 bits.
  - Le début de trame est détecté après la réception du caractère « : ».
  - Un retour chariot suivi d'un retour à la ligne marquent la fin de trame.
  - L'intégrité de la trame est vérifiée à l'aide d'un octet appelé « somme de contrôle LRC », contenu dans la trame.

## Zone Arrêt

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



La zone Arrêt vous permet d'indiquer le nombre de bits d'arrêt utilisés pour la communication. Ce champ est défini en fonction des autres équipements. Les valeurs possibles sont les suivantes :

- 1 bit
- 2 bits

## Zone Parité

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Cette zone vous permet de définir si un bit de parité est ajouté ou non, ainsi que son type : Ce champ est défini en fonction des autres équipements. Les valeurs possibles sont les suivantes :

- **Paire**
- **Impaire**
- **Aucune**

## Sous-chapitre 4.3

### Programmation d'une communication Modbus Serial

---

#### Objet de cette section

Cette partie décrit l'aspect programmation dans la mise en œuvre d'une communication Modbus Serial.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Services pris en charge par le processeur maître d'une liaison Modbus	63
Services pris en charge par le processeur esclave d'une liaison Modbus	65

## Services pris en charge par le processeur maître d'une liaison Modbus

### Fonctions de communication

Trois fonctions de communication spécifiques sont définies pour émettre et recevoir des données sur une voie de communication Modbus :

- `READ_VAR` : pour lire des variables.
- `WRITE_VAR` : pour écrire des variables.
- `DATA_EXCH` : pour envoyer des requêtes Modbus à un autre appareil avec le protocole sélectionné.

Pour plus d'informations sur ces fonctions de communication, consultez le chapitre *Informations générales sur les fonctions de communication M340 (voir EcoStruxure™ Control Expert, Communication, Bibliothèque de blocs)*.

### Echanges de données

La lecture ou l'écriture de variables s'effectue par l'envoi des requêtes ci-dessous à l'appareil esclave cible.

Ces requêtes utilisent les fonctions de communication `READ_VAR`, `WRITE_VAR` et `DATA_EXCH` :

Requête Modbus	Code fonction	Fonction de communication
Lecture de bits	16#01 ou 16#02	<code>READ_VAR</code>
Lecture de mots	16#03 ou 16#04	<code>READ_VAR</code>
Ecriture de bits	16#0F	<code>WRITE_VAR</code>
Ecriture de mots	16#10	<code>WRITE_VAR</code>
Autre requête	Tout	<code>DATA_EXCH</code>

**NOTE** : `WRITE_VAR` est utilisable en mode diffusion (mais pas `READ_VAR`). Dans ce cas, l'automate ne reçoit pas de réponse. L'envoi d'une requête de diffusion réinitialise le bit d'activité et le code 16#01 (échange interrompu à expiration du délai) est renvoyé dans le 2e mot de gestion de l'EF.

**NOTE** : l'automate Modicon M340 peut lire les objets de type `%I` et `%IW`. Dans ce cas la fonction `READ_VAR` génère une requête Modbus : FC 0x2 ou 0x4. Pour les automates Quantum, elle permet d'accéder à l'état d'entrée ou aux registres d'état d'entrée.

Plus généralement, il est possible d'envoyer n'importe quelle requête Modbus à un appareil esclave par la fonction de communication `DATA_EXCH`.

### Annulation d'un échange

Deux types de programmation permettent d'annuler un échange effectué par les fonctions de communication :

- Utilisation de la fonction `CANCEL`
- Utilisation du bit d'annulation de la fonction de communication

Pour des informations détaillées sur l'annulation d'une fonction de communication, consultez le document *EcoStruxure™ Control Expert - Communication - Bibliothèque de blocs*.



## Services pris en charge par le processeur esclave d'une liaison Modbus

### Présentation

Lorsqu'ils sont utilisés comme processeur esclave dans une liaison Modbus, les processeurs suivants prennent en charge plusieurs services :

- BMX P34 1000,
- BMX P34 2000,
- BMX P34 2010/20102,
- BMX P34 2020.

### Echanges de données

Un processeur esclave gère les requêtes suivantes :

Requête Modbus	Code fonction	Objet automate
Lecture de n bits de sortie	16#01	%M
Lecture de n bits d'entrée	16#02	%M
Lecture de n mots de sortie	16#03	%MW
Lecture de n mots d'entrée	16#04	%MW
Ecriture d'un seul bit de sortie	16#05	%M
Ecriture d'un seul mot de sortie	16#06	%MW
Ecriture de n bits de sortie	16#0F	%M
Ecriture de n mots de sortie	16#10	%MW

## Diagnostic et maintenance

Les informations de diagnostic et maintenance accessibles à partir d'une liaison Modbus sont indiquées ci-dessous :

Désignation	Code fonction / Code sous-fonction
Echo	16#08 / 16#00
Lecture des registres de diagnostic de l'automate	16#08 / 16#02
Remise à 0 des compteurs et des registres de diagnostic de l'automate	16#08 / 16#0A
Lecture du nombre de messages sur le bus	16#08 / 16#0B
Lecture du nombre d'erreurs de communication détectées sur le bus	16#08 / 16#0C
Lecture du nombre d'exceptions détectées sur le bus	16#08 / 16#0D
Lecture du nombre de messages reçus de l'esclave	16#08 / 16#0E
Lecture du nombre de non réponses de l'esclave	16#08 / 16#0F
Lecture du nombre d'accusés de réception négatifs de l'esclave	16#08 / 16#10
Lecture du nombre de réponses d'exception esclave occupé	16#08 / 16#11
Lecture du nombre de dépassements de caractères sur le bus	16#08 / 16#12
Lecture du compteur d'événements	16#0B
Lecture de l'événement connexion	16#0C
Lecture de l'identification	16#11
Lecture de l'identification de l'équipement	16#2B / 16#0E

## Sous-chapitre 4.4

### Mise au point d'une communication Modbus Serial

#### Ecran de mise au point d'une communication Modbus Série

##### Général

L'écran de mise au point d'une communication Modbus Série est accessible en mode connecté.

##### Accès à l'écran de mise au point

Le tableau suivant présente la marche à suivre pour accéder à l'écran de mise au point d'une communication Modbus Série :

Etape	Action
1	Accédez à l'écran de configuration d'une communication Modbus Série. <i>(voir page 54)</i>
2	Sélectionnez l'onglet "Mise au point" dans l'écran qui apparaît.

##### Description de l'écran Mise au point

L'écran de mise au point comprend 2 zones :

- la zone Type,
- la zone Compteurs.

##### Zone Type

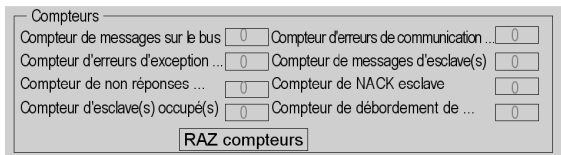
Cette zone se présente comme ceci :



Cette zone rappelle le type de fonction Modbus configuré (Maître, dans le cas présent).

## Zone Compteurs

Cette zone se présente comme ceci :



Cette zone affiche les différents compteurs de mise au point.

Le bouton RAZ Compteurs provoque la remise à 0 de tous les compteurs du mode mise au point.

## Opération de comptage

Les compteurs de mise au point d'une communication Modbus Série sont :

- Compteur de messages sur le bus : ce compteur indique le nombre de messages que le processeur a détectés sur la liaison série. Les messages dont la vérification CRC est négative ne sont pas pris en compte.
- Compteur d'erreurs de communication sur le bus : ce compteur indique le nombre de vérifications CRC négatives comptées par le processeur. Dans le cas d'une erreur détectée au niveau des caractères (dépassement, erreur de parité) ou dans le cas d'un message d'une longueur inférieure à 3 octets, le système qui reçoit les données ne peut pas effectuer la vérification CRC. Dans ces cas-là, le compteur est aussi incrémenté.
- Compteur d'erreurs d'exception d'esclaves : ce compteur indique le nombre d'erreurs d'exception Modbus détectées par le processeur.
- Compteur de messages d'esclaves : ce compteur indique le nombre de messages reçus et traités par la liaison Modbus.
- Compteur de non-réponses d'esclave(s) : ce compteur indique le nombre de messages émis par le système distant, pour lesquels il ne renvoie pas de réponse (réponse normale ou réponse d'exception). Ce compteur compte aussi le nombre de messages reçus en mode diffusion.
- Compteur d'accusés de réception négatifs d'esclaves : ce compteur indique le nombre de messages adressés au système distant pour lesquels il renvoie un accusé de réception négatif.
- Compteur d'esclaves occupés : ce compteur indique le nombre de messages adressés au système distant pour lesquels il renvoie une réponse d'exception esclave occupé.
- Compteur de dépassements de caractères sur le bus : ce compteur indique le nombre de messages envoyés au processeur et dont il ne peut pas accuser réception en raison d'un dépassement de caractères sur le bus. Ce dépassement est causé par :
  - des données de type caractères transmises sur le port série plus vite qu'elle ne peuvent être stockées,
  - une perte de données due à une anomalie matérielle.

**NOTE :** Tous les compteurs sont incrémentés à partir du dernier redémarrage, de la dernière opération d'effacement de compteurs ou de la dernière mise sous tension du processeur.

---

# Chapitre 5

## Communication Mode caractère pour les processeurs Modicon M340

---

### Objet de cette section

Ce chapitre présente la mise en œuvre logicielle de la communication Mode caractère pour les processeurs Modicon M340.

### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
5.1	Généralités	70
5.2	Configuration d'une communication en mode caractère	73
5.3	Programmation d'une communication en mode caractère	82
5.4	Mise au point d'une communication en mode caractère	84

# Sous-chapitre 5.1

## Généralités

---

### Objet de cette section

Cette partie présente les généralités sur la communication en Mode caractère et ses services.

### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
A propos de la communication Mode caractère	71
Performances	72

## A propos de la communication Mode caractère

### Présentation

La communication Mode caractère permet d'exécuter des fonctions de dialogue et de communication avec les équipements suivants :

- périphériques usuels (imprimante, écran-clavier, terminal d'atelier...),
- périphériques spécialisés (lecteur de codes-barres par exemple),
- calculateurs (contrôle, gestion de production, etc.),
- équipements hétérogènes (commandes numériques, variateurs, etc.),
- modem externe.

### AVERTISSEMENT

#### **PERTE DE DONNEES CRITIQUES**

N'utilisez les ports de communication que pour des transferts de données non critiques.

**Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

## Performances

### Présentation

Les tableaux suivants décrivent les durées d'échange typiques en Mode caractère.

Les résultats affichés correspondent à la durée moyenne de la fonction `PRINT_CHAR` en millisecondes.

### Définition de la durée d'échange

La durée d'échange est le laps de temps écoulé entre la création d'un échange et la fin de cet échange. Elle inclut le temps de communication sur la liaison série.

L'échange est créé lorsque la fonction de communication est appelée.

Il se termine lorsque l'un des événements ci-dessous survient :

- Des données sont reçues.
- Une anomalie survient.
- Le délai arrive à expiration.

### Durées d'échange

Le tableau ci-dessous indique les durées d'échange pour la transmission de 80 caractères en Mode caractère sur un processeur BMX P34 2020 avec différents débits en bauds et différentes durées de cycle :

Durée d'échange en ms		Durée du cycle en ms				
		10	20	50	100	255
<b>Débits en bauds de la communication en bits par seconde</b>	<b>1 200</b>	805	820	850	900	980
	<b>4 800</b>	210	220	250	300	425
	<b>9 600</b>	110	115	145	200	305
	<b>19 200</b>	55	60	95	100	250

Les durées d'échange sur un processeur BMX P34 2000/2010/20102 sont similaires à celles sur un processeur BMX P34 2020. Les durées d'échange sur un BMX P34 1000 sont 10 % plus rapides.

**NOTE** : toutes les durées d'échange indiquées ci-dessus proviennent de mesures présentant une marge de précision de +/- 10 ms.



---

## Sous-chapitre 5.2

### Configuration d'une communication en mode caractère

---

#### Objet de cette section

Cette partie décrit l'aspect configuration dans la mise en œuvre d'une communication en mode caractère.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Ecran de configuration d'une communication Mode caractère	74
Paramètres de détection de fin de message en mode caractère	76
Paramètres des signaux et de la ligne physique en mode caractère	78
Paramètres de la transmission Mode caractère	80

## Ecran de configuration d'une communication Mode caractère

### Général

Les pages suivantes présentent l'écran de configuration de la communication en mode caractère.

### Accès à l'écran de configuration

Le tableau suivant indique la marche à suivre pour accéder à l'écran de configuration d'une communication en mode caractère :

Etape	Action
1	Ouvrez le sous-répertoire PortSérie dans le Navigateur du projet ( <i>voir page 50</i> ).
2	Dans l'écran qui s'affiche, sélectionnez <b>Liaison mode caractère</b> dans la zone <b>Fonction</b> .

### Ecran de configuration d'une communication Mode caractère

La figure ci-dessous représente l'écran de configuration par défaut de la communication Mode caractère :

The screenshot shows the 'Configuration' dialog box for character mode communication. It is organized into several sections:

- Arrêt sur réception (Receive stop):**
  - Caractère 1 (Character 1):** Includes checkboxes for 'Arrêt' (checked), 'CR', 'LF', and 'Caractère inclus' (unchecked). A numeric field shows '0'.
  - Caractère 2 (Character 2):** Includes checkboxes for 'Arrêt' (checked), 'CR', 'LF', and 'Caractère inclus' (unchecked). A numeric field shows '0'.
- Vitesse de transmission (Transmission speed):** A dropdown menu is set to '9600 bits/s'.
- Arrêt sur silence (Silence stop):** Includes a checkbox for 'Arrêt' (checked) and a numeric field set to '2' ms.
- Données (Data):** Radio buttons for '7 bits', '8 bits' (checked), and '9 bits'.
- Arrêt (Stop):** Radio buttons for '1 bit' (checked) and '2 bits'.
- Parité (Parity):** Radio buttons for 'Paire' (unchecked), 'Impaire' (checked), and 'Aucune' (unchecked).
- Ligne physique (Physical line):** Radio buttons for 'RS232' (checked) and 'RS485'.
- Signaux (Signals):** Radio buttons for 'RX/TX' (checked), 'RX/TX + RTS/CTS DTE mode', 'RX/TX + RTS/CTS DCE mode', and 'RX/TX + RTS/CTS + DTR/DSR/DCD'.
- Retard RTS/CTS (RTS/CTS delay):** A numeric field set to '0' followed by 'X 100 ms'.

## Description

Ces zones permettent de configurer les paramètres de voie. En mode connecté, ces zones sont accessibles. En mode local, elles sont accessibles, mais les paramètres inaccessibles sont grisés.

Le tableau ci-dessous présente les différentes zones de l'écran de configuration de la communication Mode caractère :

Élément	Commentaire
Paramètres relatifs à la détection de fin de message ( <i>voir page 76</i> )	Ces paramètres sont accessibles dans 2 zones : <ul style="list-style-type: none"> <li>● Arrêt en réception</li> <li>● Arrêt sur silence</li> </ul>
Paramètres liés aux signaux et à la ligne physique ( <i>voir page 78</i> )	Ces paramètres sont accessibles dans 3 zones : <ul style="list-style-type: none"> <li>● Ligne physique</li> <li>● Signaux</li> <li>● Retard RTS/CTS</li> </ul>
Paramètres de la transmission ( <i>voir page 80</i> )	Ces paramètres sont accessibles dans 4 zones : <ul style="list-style-type: none"> <li>● Vitesse de transmission</li> <li>● Données</li> <li>● Bits d'arrêt</li> <li>● Parité</li> </ul>

## Valeurs par défaut

Le tableau ci-dessous montre les valeurs par défaut des paramètres de la communication Mode caractère :

Paramètres de configuration	Valeur	
Paramètres de détection de fin de message	<b>Arrêt en réception</b>	Aucune
	<b>Arrêt sur silence</b>	Aucune
Paramètres des signaux et de la ligne physique	<b>Ligne physique</b>	RS232
	<b>Signaux</b>	RX/TX
Paramètres de la transmission	<b>Vitesse de transmission</b>	9 600 bits/s
	<b>Données</b>	8 bits
	<b>Arrêter</b>	1 bit
	<b>Parité</b>	Impaire

## Paramètres de détection de fin de message en mode caractère

### Présentation

Les paramètres de détection de fin de message sont accessibles dans deux zones :

- Zone **Arrêt en réception** : arrêt en cas de réception d'un caractère spécial.
- Zone **Arrêt sur silence** : arrêt en cas de silence.

### Conditions d'utilisation

Le fait de sélectionner **Arrêt sur silence** désélectionne **Arrêt en réception** (et inversement).

#### NOTE :

Pour configurer une voie en mode caractère sans paramètres d'arrêt, décochez les cases **Arrêt** dans les zones de configuration suivantes :

- **Arrêt en réception** → **Caractère 1**
- **Arrêt en réception** → **Caractère 2**
- **Arrêt sur silence**

### Zone Arrêt en réception

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :

Arrêt en réception

Caractère 1

Arrêter

CR  LF

Caractères inclus

Caractère 2

Arrêter

CR  LF

Caractères inclus

Une requête de réception peut prendre fin dès qu'un caractère spécifique est reçu.

En cochant l'option **Arrêt**, il est possible de configurer l'activation de l'option **Arrêt en réception** par un caractère spécifique de fin de message :

- **CR** : permet de détecter la fin du message à l'aide d'un retour chariot.
- **LF** : permet de détecter la fin du message à l'aide d'un retour à la ligne.
- Champ de saisie de données : permet d'identifier un caractère de fin de message autre que le retour chariot ou le retour à la ligne, à l'aide d'une valeur décimale :
  - comprise entre 0 et 255 si les données sont codées sur 8 bits ;
  - comprise entre 0 et 127 si les données sont codées sur 7 bits.
- Caractère inclus : permet d'inclure le caractère de fin de message dans la table de réception de l'application automate.

Il est possible de configurer deux caractères de fin de réception. Dans la fenêtre ci-dessus, la fin de réception d'un message est détectée par un caractère de retour à la ligne ou de retour chariot.

## Zone Arrêt sur silence

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :

Cette zone vous permet de détecter la fin d'un message reçu, par l'absence de caractères de fin de message pendant un laps de temps donné.

La fonction **Arrêt sur silence** est activée lorsque vous cochez la case **Arrêter**. La durée du silence (exprimée en millisecondes) est définie dans le champ de saisie de données.

La valeur minimale de cette durée correspond à la transmission de 1,5 caractère. Exprimée en nombre de bits et selon la configuration des bits de début et d'arrêt, la durée minimale du silence est la suivante :

Longueur totale des caractères (bits)	Durée minimale du silence (bits)
8	12
9	12
10	15
11	15

Convertissez le nombre dans la colonne de droite en temps selon la vitesse de transmission configurée.

**NOTE** : les valeurs disponibles vont de 1 à 10 000 ms et dépendent de la vitesse de transmission sélectionnée.

## Paramètres des signaux et de la ligne physique en mode caractère

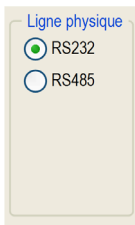
### Présentation

Les paramètres des signaux et de la ligne physique sont accessibles dans trois zones :

- **Ligne physique**
- **Signaux**
- **Retard RTS/CTS**

### Zone Ligne physique

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :

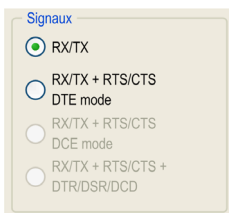


Dans cette zone, vous pouvez choisir entre deux types de ligne physique pour le port série des processeurs BMX P34 1000/2000/2010/20102/2020 :

- la ligne RS232,
- la ligne RS485.

### Zone Signaux

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Dans cette zone, vous pouvez sélectionner les signaux pris en charge par la ligne physique RS232 :

- **RX/TX**
- **RX/TX + RTS/CTS DTE mode**

Si la ligne RS485 est configurée, la zone est entièrement grisée et la valeur par défaut est **RX/TX**.

**NOTE** : seuls les signaux **RX/TX** et **RX/TX + RTS/CTS DTE mode** sont disponibles si vous configurez le port série en Mode caractère.

## Zone Retard RTS/CTS

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



La zone **Retard RTS/CTS** n'est disponible que lorsque les cases RS232 et **RX/TX+RTS/CTS DTE mode** sont cochées.

Un algorithme de contrôle de flux RTS/CTS est sélectionné : avant de transmettre une chaîne de caractères, le système attend que le signal CTS (Clear To Send) soit activé. Cette zone vous permet de saisir le temps d'attente maximum entre deux signaux. Lorsque ce délai est écoulé, la requête n'est pas transmise sur le bus. Les valeurs configurables vont de 0 à 10 s.

**NOTE** : la valeur par défaut est 0 ms.

**NOTE** : une valeur de 0 s indique que le retard entre deux signaux n'a pas été pris en charge.

## Algorithme de contrôle de flux RTS/CTS

L'objectif est d'éviter le dépassement du tampon de réception.

Le signal de sortie RTS de chaque équipement est connecté au signal d'entrée CTS de l'autre équipement. L'émetteur (M340) est autorisé à émettre des données dès qu'il reçoit le signal d'entrée RTS (par exemple, un autre M340) sur son entrée CTS. Cet algorithme est symétrique et permet une communication asynchrone en duplex intégral.

## Paramètres de la transmission Mode caractère

### Présentation

Les paramètres de transmission sont accessibles dans quatre zones :

- **Vitesse de transmission**
- **Données**
- **Arrêt**
- **Parité**

### Zone Vitesse de transmission

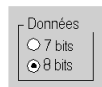
Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Vous pouvez utiliser cette zone pour sélectionner la vitesse de transmission du protocole Mode caractère. La vitesse sélectionnée doit être cohérente avec les autres équipements. Les valeurs possibles sont 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200 et 38 400 bits par seconde.

### Zone Données

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Dans cette zone, vous pouvez spécifier la taille des données échangées sur la liaison. Les valeurs disponibles sont les suivantes :

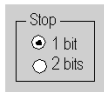
- 7 bits
- 8 bits

Il est recommandé de régler le nombre de bits de données en fonction de l'équipement distant utilisé.



## Zone Arrêt

Cette zone se présente comme ceci :



La zone **Arrêt** vous permet d'indiquer le nombre de bits d'arrêt utilisés pour la communication. Il est recommandé de régler le nombre de bits d'arrêt en fonction de l'équipement distant utilisé.

Les valeurs possibles sont les suivantes :

- 1 bit
- 2 bits

## Zone Parité

Cette zone de configuration apparaît à l'écran, comme indiqué ci-dessous :



Cette zone vous permet de définir si un bit de parité est ajouté ou non, ainsi que son type. Il est recommandé de régler la parité en fonction de l'équipement distant utilisé. Les valeurs possibles sont les suivantes :

- **Paire**
- **Impaire**
- **Aucune**

## Sous-chapitre 5.3

### Programmation d'une communication en mode caractère

#### Fonctions de communication Mode caractère

##### Fonctions disponibles

Trois fonctions de communication spécifiques permettent d'émettre et recevoir des données via une voie de communication en mode caractère :

- `PRINT_CHAR` : envoi d'une chaîne de caractères de 1 024 octets maximum.
- `INPUT_CHAR` : lecture d'une chaîne de caractères de 1 024 octets maximum.
- `INPUT_BYTE` : lecture d'un tableau de 1 024 octets maximum.

Pour plus d'informations sur les fonctions de communication, consultez le chapitre *Informations générales sur les fonctions de communication M340 (voir EcoStruxure™ Control Expert, Communication, Bibliothèque de blocs)*.

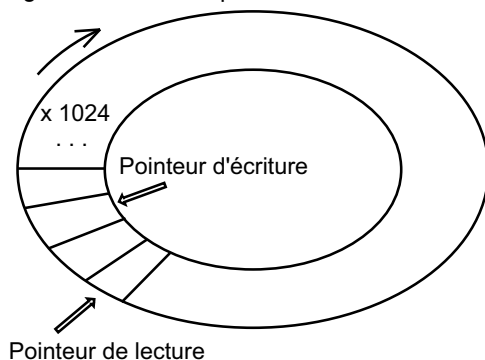
**NOTE** : pour confirmer le bit d'activité de la fonction `INPUT_CHAR`, il est nécessaire de définir un délai d'expiration si la voie est configurée sans arrêt en cas de silence. Pour la fonction `PRINT_CHAR`, la définition d'un délai d'expiration est facultative mais conseillée.

##### Mécanisme interne de l'UC

Le port série de l'automate Modicon M340 est en duplex intégral. Par conséquent, une fonction `PRINT_CHAR` peut être envoyée même si une fonction `INPUT_CHAR` ou `INPUT_BYTE` a été envoyée ou est en attente d'envoi.

L'accès en lecture et écriture s'effectue grâce à deux pointeurs indépendants.

La figure ci-dessous représente ce mécanisme :



Les données reçues sont stockées dans un tampon cyclique de 1 024 bits. Une fois le tampon saturé, le 1 025e bit reçu remplace le 1er bit, et ainsi de suite. Chaque bit du tampon lu à l'aide de la fonction `INPUT_CHAR` est réinitialisé.

L'UC enregistre l'écho des données émises dans le même tampon que les données reçues. Il est donc obligatoire de vider le tampon de l'UC après chaque `PRINT_CHAR` ou avant que quelqu'un n'envoie des données à la voie. Sinon, les données reçues suite à une fonction `INPUT_CHAR` ou `INPUT_BYTE` ne seront pas les bonnes.

Pour vider le tampon de l'UC, vous pouvez mettre à 1 le paramètre d'entrée `RAZ` de la fonction de lecture et annuler cette dernière avant l'expiration du délai. Le tampon est d'abord réinitialisé, puis le processeur attend la réception des données.

**NOTE** : L'utilisation de cette fonction est conseillée pour démarrer correctement une réception en éliminant les anciennes données pouvant rester dans le tampon.

### Annulation d'un échange

Deux types de programmation permettent d'annuler un échange effectué par les fonctions de communication :

- Utilisation de la fonction `CANCEL`
- Utilisation du bit d'annulation de la fonction de communication

Pour des informations détaillées sur l'annulation d'une fonction de communication, consultez le document *EcoStruxure™ Control Expert - Communication - Bibliothèque de blocs*.

## Sous-chapitre 5.4

### Mise au point d'une communication en mode caractère

#### Ecran de mise au point d'une communication Mode caractère

##### Généralités

L'écran de mise au point du Mode caractère est accessible en mode connecté.

##### Accès à l'écran de mise au point

Le tableau suivant présente la marche à suivre pour accéder à l'écran de mise au point de la communication en Mode caractère :

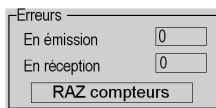
Etape	Action
1	Accédez à l'écran de configuration de la communication en Mode caractère. <i>(voir page 74)</i>
2	Sélectionnez l'onglet "Mise au point" dans l'écran qui apparaît.

##### Description de l'écran Mise au point

L'écran de mise au point est composé d'une zone Erreurs et une zone Signaux.

##### Zone Erreurs

La zone Erreurs se présente comme suit :



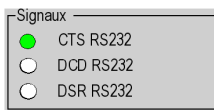
Cette zone indique le nombre d'interruptions de communication comptabilisées par le processeur :

- **En émission** : nombre d'interruptions en émission (image du mot %MW4).
- **En réception** : nombre d'interruptions en réception (image du mot %MW5).

Le bouton RAZ compteurs provoque la remise à 0 des deux compteurs.

## Zone Signaux

La zone Signaux se présente comme suit :



Cette zone indique l'activité des signaux :

- **CTS RS232** : indique l'activité du signal CTS.
- **DCD RS232** : non géré par le processeur (aucune activité sur ce voyant).
- **DSR RS232** : non géré par le processeur (aucune activité pour ce voyant).



---

# Chapitre 6

## Objets langage des communications Modbus et Mode caractère

---

### Objet de ce chapitre

Ce chapitre décrit les objets langage associés aux communications Modbus et Mode caractère ainsi que les différents moyens de les utiliser.

### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
6.1	Objets langage et IODDT des communications Modbus et Mode caractère	88
6.2	Objets langage et IODDT génériques pour les protocoles de communication	96
6.3	Objets langage et IODDT associés aux communications Modbus	100
6.4	Objets langage et IODDT associés aux communications Modbus en mode caractère	108
6.5	Type d'IODDT Type T_GEN_MOD applicable à tous les modules	116

## Sous-chapitre 6.1

### Objets langage et IODDT des communications Modbus et Mode caractère

---

#### Objet de cette partie

Cette partie présente les généralités des objets langage et IODDT des communications Modbus et Mode caractère.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Présentation des objets langage pour les communications Modbus et Mode caractère	89
Objets langage à échanges implicites associés à la fonction métier	90
Objets langage à échanges explicites associés à la fonction métier	91
Gestion des échanges et des comptes rendus avec des objets explicites	93



## Présentation des objets langage pour les communications Modbus et Mode caractère

### Généralités

Les IODDT sont prédéfinis par le constructeur. Ils contiennent des objets langage d'E/S appartenant à une voie d'un module métier.

Les communications Modbus et Mode caractère ont trois IODDT associés :

- T\_COM\_STS\_GEN, applicable à tous les protocoles de communication, sauf Fipio et Ethernet ;
- T\_COM\_MBP, réservé à la communication Modbus ;
- T\_COM\_CHAR\_BMX, réservé à la communication Mode caractère.

**NOTE** : les variables IODDT peuvent être créées de deux façons :

- dans l'onglet Objets d'E/S (*voir EcoStruxure™ Control Expert, Modes de fonctionnement*) ;
- dans l'éditeur de données (*voir EcoStruxure™ Control Expert, Modes de fonctionnement*).

### Types d'objets langage

Chaque IODDT contient un ensemble d'objets langage permettant de les commander et de vérifier leur bon fonctionnement.

Il existe deux types d'objets langage :

- Objets à échanges implicites : ces objets sont échangés automatiquement à chaque cycle de la tâche associée au module.
- Objets à échanges explicites : ces objets sont échangés à la demande de l'application, à l'aide d'instructions d'échange explicite.

Les échanges implicites concernent l'état des processeurs, les signaux de communication, les esclaves, etc.

Les échanges explicites permettent de définir les paramètres du processeur et d'effectuer des diagnostics.

## Objets langage à échanges implicites associés à la fonction métier

### Présentation

Une interface métier intégrée, ou l'ajout d'un module, enrichit automatiquement le projet d'objets langage permettant de programmer cette interface ou ce module.

Ces objets correspondent aux images des entrées/sorties et informations logicielles du module ou de l'interface intégrée métier.

### Rappels

Les entrées (%I et %IW) du module sont mises à jour dans la mémoire automate en début de tâche, ou quand l'automate est en mode RUN ou STOP.

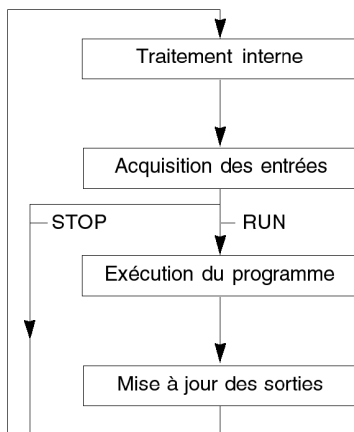
Les sorties (%Q et %QW) sont mises à jour à la fin de la tâche, uniquement lorsque l'automate est en mode RUN.

**NOTE** : Lorsque la tâche est en mode STOP et selon la configuration choisie :

- les sorties sont mises en position de repli (mode de repli) ;
- les sorties sont maintenues à leur dernière valeur (mode maintien).

### Illustration

Le schéma ci-dessous illustre le cycle de fonctionnement d'une tâche automate (exécution cyclique) :



## Objets langage à échanges explicites associés à la fonction métier

### Présentation

Les échanges explicites sont des échanges effectués sur demande du programme utilisateur à l'aide des instructions ci-dessous :

- `READ_STS` (voir *EcoStruxure™ Control Expert, Gestion des E/S, Bibliothèque de blocs*) : lecture des mots d'état
- `WRITE_CMD` (voir *EcoStruxure™ Control Expert, Gestion des E/S, Bibliothèque de blocs*) : écriture des mots de commande

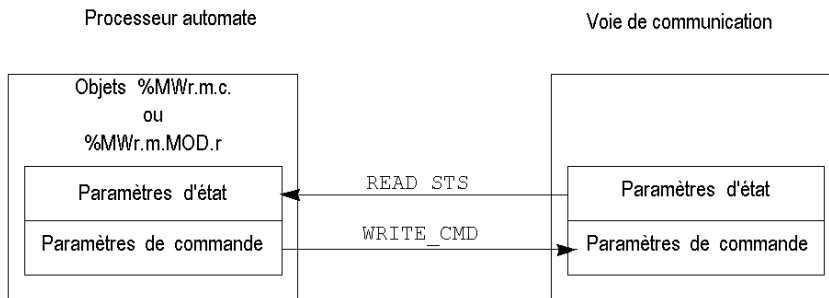
Ces échanges s'appliquent à un ensemble d'objets `%MW` de même type (état, commande ou paramètre) appartenant à une voie.

**NOTE** : Ces objets fournissent des informations sur le processeur ou le module, et peuvent permettre de les commander (par exemple : commande des bascules) et de définir ses modes de fonctionnement (sauvegarde et restauration des paramètres de réglage en cours d'application).

**NOTE** : Les instructions `READ_STS` et `WRITE_CMD` sont exécutées en même temps que la tâche qui les appelle et toujours correctement. Le résultat de ces instructions est disponible juste après leur exécution.

### Principe général d'utilisation des instructions explicites

Le schéma ci-dessous présente les différents types d'échanges explicites possibles entre le processeur et la voie de communication :



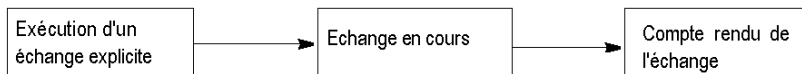
## Gestion des échanges

Lors d'un échange explicite, il faut contrôler le déroulement de celui-ci, afin de ne prendre en compte les données que lorsque l'échange a été correctement effectué.

Pour cela, 2 types d'information sont disponibles :

- les informations relatives à l'échange en cours (*voir EcoStruxure™ Control Expert, Gestion des E/S, Bibliothèque de blocs*) ;
- le compte rendu de l'échange (*voir EcoStruxure™ Control Expert, Gestion des E/S, Bibliothèque de blocs*).

Le synoptique ci-dessous décrit le principe de gestion d'un échange :



**NOTE** : afin d'éviter plusieurs échanges explicites simultanés pour la même voie, il est nécessaire de tester la valeur du mot EXCH\_STS (%MWx.m.c.0) de l'IODDT associé à la voie avant d'appeler une fonction élémentaire (EF) utilisant cette voie.

## Gestion des échanges et des comptes rendus avec des objets explicites

### Vue d'ensemble

Lorsque les données sont échangées entre la mémoire automate et le module, la prise en compte par le coupleur peut nécessiter plusieurs cycles de tâche.

Pour gérer les échanges, tous les IODDT utilisent deux mots :

- EXCH\_STS (%MWr.m.c.0) : échange en cours.
- EXCH\_RPT (%MWr.m.c.1) : compte rendu.

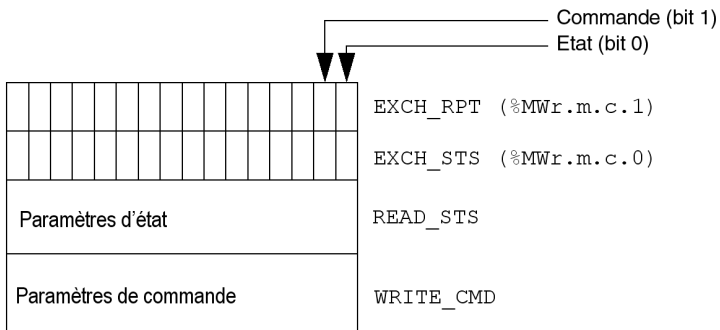
#### NOTE :

selon la localisation du module, la gestion des échanges explicites (%MW0.0.MOD.0.0 par exemple) ne sera pas détectée par l'application :

- Pour les modules en rack, les échanges explicites ont lieu immédiatement sur le bus automate local et se terminent avant la fin de la tâche d'exécution, afin que le READ\_STS, par exemple, soit toujours terminé quand le bit %MW0.0.mod.0.0 est vérifié par l'application.
- sur un bus distant (Fipio par exemple), les échanges explicites ne sont pas synchronisés avec la tâche d'exécution, donc la détection pour l'application est possible.

### Illustration

L'illustration ci-dessous présente les différents bits significatifs pour la gestion des échanges :



## Description des bits significatifs

Chacun des bits des mots EXCH\_STS (%MW<sub>r.m.c.0</sub>) et EXCH\_RPT (%MW<sub>r.m.c.1</sub>) est associé à un type de paramètre :

- Les bits de rang 0 sont associés aux paramètres d'état :
  - le bit STS\_IN\_PROGR (%MW<sub>r.m.c.0.0</sub>) indique si une demande de lecture des mots d'état est en cours.
  - Le bit STS\_ERR (%MW<sub>r.m.c.1.0</sub>) précise si une demande de lecture des mots d'état est acceptée par la voie du module.
- Les bits de rang 1 sont associés aux paramètres de commande :
  - le bit CMD\_IN\_PROGR (%MW<sub>r.m.c.0.1</sub>) indique si des paramètres de commande sont envoyés à la voie du module,
  - le bit CMD\_ERR (%MW<sub>r.m.c.1.1</sub>) indique si les paramètres de commandes sont acceptés par la voie du module.

**NOTE** : r représente le numéro du rack, m représente la position du module dans le rack, c représente le numéro de voie dans le module.

**NOTE** : les mots d'échange et de compte-rendu existent aussi au niveau des modules EXCH\_STS (%MW<sub>r.m.MOD.0</sub>) et EXCH\_RPT (%MW<sub>r.m.MOD.1</sub>) comme pour les IODDT de type T\_GEN\_MOD.

## Indicateurs d'exécution d'un échange explicite : EXCH\_STS

Le tableau ci-dessous présente les bits de contrôle des échanges explicites du mot EXCH\_STS (%MW<sub>r.m.c.0</sub>) :

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	R	Lecture des mots d'état de la voie en cours.	%MW <sub>r.m.c.0.0</sub>
CMD_IN_PROGR	BOOL	R	Echange de paramètres de commande en cours	%MW <sub>r.m.c.0.1</sub>
ADJ_IN_PROGR	BOOL	R	Echange de paramètres de réglage en cours	%MW <sub>r.m.c.0.2</sub>
RECONF_IN_PROGR	BOOL	R	Reconfiguration du module en cours	%MW <sub>r.m.c.0.15</sub>

**NOTE** : si le module n'est pas présent ou est déconnecté, les échanges utilisant des objets explicites (READ\_STS par exemple) ne sont pas envoyés au processeur (STS\_IN\_PROG (%MW<sub>r.m.c.0.0</sub>) = 0), mais les mots sont actualisés.

**Compte rendu d'échanges explicites : EXCH\_RPT**

Le tableau ci-dessous présente les bits de compte-rendu du mot EXCH\_RPT (%MWr.m.c.1) :

Symbole standard	Type	Accès	Signification	Adresse
STS_ERR	BOOL	R	Erreur détectée de lecture des mots d'état de la voie (1 = échec détecté)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Erreur détectée lors d'un échange de paramètres de commande (1 = échec détecté)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Interruptions lors d'un échange de paramètres de réglage (1 = échec détecté)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Interruptions lors de la reconfiguration de la voie (1 = échec détecté)	%MWr.m.c.1.15

## Sous-chapitre 6.2

### Objets langage et IODDT génériques pour les protocoles de communication

---

#### Objet de ce sous-chapitre

Ce sous-chapitre présente les objets langage et les IODDT génériques qui s'appliquent à tous les protocoles de communication, sauf Fipio et Ethernet.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Informations détaillées sur les objets à échange implicite IODDT de type T_COM_STS_GEN	97
Détails sur les objets à échange explicite de l'IODDT de type T_COM_STS_GEN	98



## Informations détaillées sur les objets à échange implicite IODDT de type T\_COM\_STS\_GEN

### Présentation

Le tableau suivant présente les objets à échange implicite d'IODDT de type T\_COM\_STS\_GEN qui s'appliquent à tous les protocoles de communication sauf Fipio.

### Bit d'erreur

Le tableau ci-dessous présente la signification du bit d'erreur CH\_ERROR (%l.r.m.c.ERR) :

Symbole standard	Type	Accès	Signification	Adresse
CH_ERROR	EBOOL	L	Bit d'erreur de voie de communication.	%l.r.m.c.ERR

## Détails sur les objets à échange explicite de l'IODDT de type T\_COM\_STS\_GEN

### Présentation

Cette section présente les objets à échange explicite de l'IODDT de type T\_COM\_STS\_GEN, qui s'appliquent à tous les protocoles de communication, sauf Fipio et Ethernet. Elle regroupe les objets de type mot, dont les bits ont une signification particulière. Ces objets sont présentés en détail ci-dessous.

Dans cette partie, la variable IODDT\_VAR1 est de type T\_COM\_STS\_GEN.

### Remarques

De manière générale la signification des bits est donnée pour l'état 1. Dans les cas spécifiques, chaque état du bit est expliqué.

Tous les bits ne sont pas utilisés.

### Indicateurs d'exécution d'un échange explicite : EXCH\_STS

Le tableau ci-dessous présente la signification des bits de contrôle d'échange de la voie EXCH\_STS (%MW<sub>r.m.c.0</sub>) :

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	R	Lecture des mots d'état de la voie en cours.	%MW <sub>r.m.c.0.0</sub>
CMD_IN_PROGR	BOOL	R	Echange de paramètres de commande en cours.	%MW <sub>r.m.c.0.1</sub>

### Compte rendu d'échanges explicites : EXCH\_RPT

Le tableau ci-dessous présente la signification des bits de compte rendu d'échange EXCH\_RPT (%MW<sub>r.m.c.1</sub>) :

Symbole standard	Type	Accès	Signification	Adresse
STS_ERR	BOOL	R	Erreur détectée de lecture des mots d'état de la voie.	%MW <sub>r.m.c.1.0</sub>
CMD_ERR	BOOL	R	Erreur détectée lors d'un échange de paramètres de commande.	%MW <sub>r.m.c.1.1</sub>

**Défauts standard voie : CH\_FLT**

Le tableau ci-dessous présente la signification des bits du mot d'état CH\_FLT (%MWr.m.c.2) :

Symbole standard	Type	Accès	Signification	Adresse
NO_DEVICE	BOOL	R	Aucun équipement ne fonctionne sur la voie.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	R	Un équipement sur la voie n'est pas fonctionnel.	%MWr.m.c.2.1
BLK	BOOL	R	Bornier non connecté.	%MWr.m.c.2.2
TO_ERR	BOOL	R	Délai d'attente dépassé (analyse nécessaire).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Erreur interne détectée ou autotest de la voie.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Configurations matérielle et logicielle différentes.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Analyse de communication nécessaire sur la voie.	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Erreur d'application détectée (erreur de réglage ou de configuration).	%MWr.m.c.2.7

La lecture est effectuée par l'instruction READ\_STS (IODDT\_VAR1).

## Sous-chapitre 6.3

### Objets langage et IODDT associés aux communications Modbus

---

#### Objet de cette partie

Cette partie présente les objets langage et l'IODDT qui sont associés à la fonction de communication Modbus.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Informations détaillées sur les objets langage à échange explicite pour une fonction Modbus	101
Informations détaillées sur les objets à échanges explicites des IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT	102
Informations détaillées sur les objets à échanges explicites IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT	103
Informations détaillées sur les objets langage associés au mode de communication Modbus	106

## Informations détaillées sur les objets langage à échange explicite pour une fonction Modbus

### Présentation

Le tableau ci-dessous présente les objets langage pour une communication Modbus maître ou esclave. Ces objets ne sont pas intégrés dans les IODDT.

### Liste des objets à échanges explicites en mode maître ou esclave

Le tableau ci-dessous présente les objets à échange explicite :

Adresse	Type	Accès	Signification
%MW $r$ .m.c.4	INT	R	Nombre de réponses reçues correctement.
%MW $r$ .m.c.5	INT	R	Nombre de réponses reçues avec erreur CRC.
%MW $r$ .m.c.6	INT	R	Nombre de réponses reçues avec un code d'exception en mode esclave.
%MW $r$ .m.c.7	INT	R	Nombre de messages émis en mode esclave.
%MW $r$ .m.c.8	INT	R	Nombre de messages émis sans réponse en mode esclave.
%MW $r$ .m.c.9	INT	R	Nombre de réceptions avec accusé de réception négatif.
%MW $r$ .m.c.10	INT	R	Nombre de messages réitérés en mode esclave.
%MW $r$ .m.c.11	INT	R	Nombre d'erreurs de caractères détectées.
%MW $r$ .m.c.24.0	BOOL	RW	Réinitialisation des compteurs d'erreurs détectées.

## Informations détaillées sur les objets à échanges explicites des IODDT de types T\_COM\_MB\_BMX et T\_COM\_MB\_BMX\_CONF\_EXT

### Présentation

Les tableaux ci-dessous montrent les objets à échanges implicites des IODDT de types T\_COM\_MB\_BMX et T\_COM\_MB\_BMX\_CONF\_EXT, qui sont applicables aux communications série Modbus. Ils diffèrent en termes de **disponibilité des objets de configuration** (voir page 105).

### Bit CH\_ERROR

Le tableau ci-dessous indique la signification du bit d'erreur CH\_ERROR (%I<sub>r.m.c</sub>.ERR) :

Symbole standard	Type	Accès	Signification	Adresse
CH_ERROR	EBOOL	L	Bit d'erreur détectée de voie de communication	%I <sub>r.m.c</sub> .ERR

### Objet mot en mode Maître Modbus

Le tableau ci-dessous indique la signification du bit du mot INPUT\_SIGNALS (%I<sub>Wr.m.c</sub>.0) :

Symbole standard	Type	Accès	Signification	Adresse
DCD	BOOL	L	Signal RS232 DCD (applicable seulement au module BMX NOM 0200).	%I <sub>Wr.m.c</sub> .0.0
CTS	BOOL	L	Signal RS232 CTS	%I <sub>Wr.m.c</sub> .0.2
DSR	BOOL	L	Signal RS232 DSR (applicable seulement au module BMX NOM 0200)	%I <sub>Wr.m.c</sub> .0.3

**NOTE** : %I<sub>Wr.m.c</sub>.0.2 est à 1 lorsque la tension sur le signal CTS est positive. Il s'applique également aux signaux DCD et DSR.

### Objet mot en mode Esclave Modbus

Les objets langage sont identiques à ceux de la fonction du maître Modbus. Seuls les objets dans le tableau ci-dessus diffèrent.

Le tableau ci-dessous indique la signification du bit du mot INPUT\_SIGNALS (%I<sub>Wr.m.c</sub>.0) :

Symbole standard	Type	Accès	Signification	Adresse
LISTEN_ONLY	BOOL	L	Mode écoute seule	%I <sub>Wr.m.c</sub> .0.8

## Informations détaillées sur les objets à échanges explicites IODDT de types T\_COM\_MB\_BMX et T\_COM\_MB\_BMX\_CONF\_EXT

### Présentation

Cette partie présente les objets à échanges explicites des IODDT de types T\_COM\_MB\_BMX et T\_COM\_MB\_BMX\_CONF\_EXT applicables au protocole Modbus série mais différent en ce qui concerne la **disponibilité des objets de configuration** (voir page 105). Elle inclut les objets de type mot, dont les bits ont une signification particulière. Ces objets sont présentés en détail ci-dessous.

Dans cette partie, la variable IODDT\_VAR1 est de type T\_COM\_STS\_GEN.

### Remarques

De manière générale, la signification des bits est donnée pour l'état 1 de ce bit. Dans les cas spécifiques chaque état du bit est expliqué.

Tous les bits ne sont pas utilisés.

### Indicateurs d'exécution d'échange explicite : EXCH\_STS

Le tableau ci-dessous présente la signification des bits de contrôle d'échange de la voie EXCH\_STS (%MWr.m.c.0) :

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	L	Lecture des mots d'état de la voie en cours.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	L	Echange de paramètres de commande en cours.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	L	Echange de paramètres de réglage en cours (non applicable au module BMX NOM 0200).	%MWr.m.c.0.2

### Compte rendu d'échange explicite : EXCH\_RPT

Le tableau ci-dessous présente la signification des bits de compte rendu d'échange EXCH\_RPT (%MWr.m.c.1) :

Symbole standard	Type	Accès	Signification	Adresse
STS_ERR	BOOL	L	Détection d'erreur de lecture des mots d'état de la voie.	%MWr.m.c.1.0
CMD_ERR	BOOL	L	Anomalie lors d'un échange de paramètres de commande.	%MWr.m.c.1.1
ADJ_ERR	BOOL	L	Anomalie lors de l'échange de paramètres de réglage en cours (non applicable au module BMX NOM 0200).	%MWr.m.c.1.2

**Détection de défauts standard de voie : CH\_FLT**

Le tableau ci-dessous présente la signification des bits du mot d'état CH\_FLT (%MWr.m.c.2) :

Symbole standard	Type	Accès	Signification	Adresse
NO_DEVICE	BOOL	L	Aucun équipement ne fonctionne sur la voie.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	L	Un équipement sur la voie n'est pas fonctionnel.	%MWr.m.c.2.1
BLK	BOOL	L	Bornier non connecté.	%MWr.m.c.2.2
TO_ERR	BOOL	L	Délai d'attente dépassé (analyse nécessaire).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	L	Erreur interne détectée ou autotest de la voie.	%MWr.m.c.2.4
CONF_FLT	BOOL	L	Configurations matérielle et logicielle différentes.	%MWr.m.c.2.5
COM_FLT	BOOL	L	Analyse de communication nécessaire sur la voie.	%MWr.m.c.2.6
APPLI_FLT	BOOL	L	Erreur d'application détectée (erreur de réglage ou de configuration).	%MWr.m.c.2.7

La lecture est effectuée par l'instruction READ\_STS (IODDT\_VAR1).

**Etat de voie spécifique : %MWr.m.c.3**

Le tableau ci-dessous présente la signification des bits du mot d'état de voie PROTOCOL (%MWr.m.c.3) :

Symbole standard	Type	Accès	Signification	Adresse
PROTOCOL	INT	R	Octet 0 = 16#06 pour la fonction Modbus maître. Octet 0 = 16#07 pour la fonction Modbus esclave. Octet 0 = 16#03 pour le mode caractère.	%MWr.m.c.3

La lecture est effectuée par l'instruction READ\_STS (IODDT\_VAR1).



**Commande de voie : %MWr.m.c.24**

Le tableau ci-dessous présente les significations des bits du mot CONTROL(%MWr.m.c.24) :

Symbole standard	Type	Accès	Signification	Adresse
DTR_ON	BOOL	L/E	Active le signal terminal de données prêt (DTR, Data Terminal Ready).	%MWr.m.c.24.8
DTR_OFF	BOOL	L/E	Réinitialise le signal terminal de données prêt (DTR, Data Terminal Ready).	%MWr.m.c.24.9
TO_MODBUS_MASTER	BOOL	L/E	Changement du mode caractère ou Modbus esclave au mode Modbus maître.	%MWr.m.c.24.12
TO_MODBUS_SLAVE	BOOL	L/E	Changement du mode caractère ou Modbus maître au mode Modbus esclave.	%MWr.m.c.24.13
TO_CHAR_MODE	BOOL	L/E	Changement du mode Modbus au mode caractère.	%MWr.m.c.24.14

La commande est exécutée par l'instruction WRITE\_CMD (IODDT\_VAR1).

Pour plus d'informations sur la modification des protocoles, consultez **Modification des protocoles** (voir page 119).

**Objets de configuration externes de type T\_COM\_MB\_BMX\_CONF\_EXT: %MWr.m.c.24.7 et %MWr.m.c.25**

Le tableau ci-dessous présente la signification du bit CONTROL (%MWr.m.c.24.7) et du mot CONTROL\_DATA (%MWr.m.c.25) spécifiquement destinés à la programmation du module BMX NOM 0200 :

Symbole standard	Type	Accès	Signification	Adresse
SAVE_SLAVE_ADDR	BOOL	L/E	Enregistre les données de commande dans la mémoire FLASH	%MWr.m.c.24.7
SLAVE_ADDR	INT	R/W	Adresse Modbus esclave à stocker dans la mémoire FLASH, de 0 à 248 (0 pour l'esclave). <b>NOTE</b> : Notez que cette fonctionnalité est facultative et qu'il est préférable de ne pas l'utiliser de manière intensive. Comme la technologie utilisée est FLASH, cela peut endommager la puce.	%MWr.m.c.25

## Informations détaillées sur les objets langage associés au mode de communication Modbus

### Présentation

Les tableaux suivants présentent tous les objets langage de configuration pour le mode de communication Modbus. Ces objets ne sont pas intégrés aux IODDT et peuvent être affichés par le programme d'application.

### Liste des objets à échanges explicites en mode maître

Le tableau ci-dessous répertorie les objets à échanges explicites.

Adresse	Type	Accès	Signification
%KWr.m.c.0	INT	R	L'octet 0 de ce mot correspond au type : <ul style="list-style-type: none"> <li>● La valeur 6 fait référence au maître.</li> <li>● La valeur 7 fait référence à l'esclave.</li> </ul>
%KWr.m.c.1	INT	R	L'octet 0 de ce mot correspond à la vitesse de transmission. Il peut avoir différentes valeurs : <ul style="list-style-type: none"> <li>● La valeur -2 (0xFE) correspond à 300 bits/s.</li> <li>● La valeur -1 (0xFF) correspond à 600 bits/s.</li> <li>● La valeur 0 (0x00) correspond à 1200 bits/s.</li> <li>● La valeur 1 (0x01) correspond à 2400 bits/s.</li> <li>● La valeur 2 (0x02) correspond à 4800 bits/s.</li> <li>● La valeur 3 (0x03) correspond à 9600 bits/s.</li> <li>● La valeur 4 (0x04) correspond à 19200 bits/s (valeur par défaut).</li> <li>● La valeur 5 (0x05) correspond à 38400 bits/s.</li> <li>● La valeur 6 (0x06) correspond à 57600 bits/s (applicable au module BMX NOM 0200 seulement)</li> <li>● La valeur 7 (0x07) correspond à 115200 bits/s (applicable au module BMX NOM 0200 seulement)</li> </ul> L'octet 1 de ce mot correspond au format : <ul style="list-style-type: none"> <li>● Bit 8 : nombre de bits (1 = 8 bits (RTU), 0 = 7 bits (ASCII))</li> <li>● bit 9 = 1 : gestion de la parité (1 = avec, 0 = sans)</li> <li>● Bit 10 : type de parité (1 = impair, 0 = pair)</li> <li>● Bit 11 : nombre de bits d'arrêt (1 = 1 bit, 0 = 2 bits)</li> <li>● Bit 13 : ligne physique (1 = RS232, 0 = RS485)</li> <li>● Bit 14 : Les signaux de modem DTR/DSR/DCD (applicable seulement au module BMX NOM 0200 et pour une ligne physique RS232). Si ce bit est mis à 1, les signaux de modem sont gérés.</li> <li>● Bit 15 : Signaux de gestion de contrôle de flux matériel RTS/CTS Si RS232 est sélectionné, 2 valeurs peuvent être associées à ce bit : 0 pour RX/TX et 1 pour RX/TX + RTS/CTS. Si RS485 est sélectionné, la valeur par défaut est 0 (RX/TX).</li> </ul>

Adresse	Type	Accès	Signification
%KWr.m.c.2	INT	R	Temporisation entre trames (en mode RTU seulement) valeur comprise entre 2 et 10 000 ms (suivant la vitesse de transmission et le format sélectionnés). Si l'option par défaut est activée, la valeur par défaut est 2 ms. 10 s correspond à un délai d'attente illimité.
%KWr.m.c.3	INT	R	En mode Modbus maître, cet objet fait référence au délai de réponse exprimé en millisecondes (entre 10 et 1 000 ms). La valeur par défaut est 100 ms. 10 s correspond à un délai d'attente illimité.
%KWr.m.c.4	INT	R	Disponible uniquement en mode Modbus maître. L'octet 0 de ce mot indique le nombre de répétitions (de 0 à 15). La valeur par défaut est 3.
%KWr.m.c.5	INT	R	Si RS232 est sélectionné ce mot correspond à la temporisation RTS/CTS exprimée en millisecondes (entre 0 et 100). Si RS485 est sélectionné, la valeur par défaut est 0.

### Liste des objets à échanges explicites en mode esclave

Les objets langage de la fonction Modbus esclave sont identiques à ceux de la fonction Modbus maître. La seule différence concerne les objets suivants :

Adresse	Type	Accès	Signification
%KWr.m.c.3	INT	R	En mode Modbus esclave, l'octet 0 de cet objet correspond au numéro de l'esclave [0/1, 247]. Pour le module BMX NOM 0200, la valeur 0 signifie que le numéro d'esclave est codé dans la mémoire FLASH
%KWr.m.c.4	INT	R	Utilisé uniquement en mode Modbus maître.

## Sous-chapitre 6.4

### Objets langage et IODDT associés aux communications Modbus en mode caractère

---

#### Objet de cette partie

Cette partie présente les objets langage et l'IODDT qui sont associés à la fonction de communication Mode caractère.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Informations détaillées sur les objets langage à échange explicite pour la communication en Mode caractère.	109
Informations détaillées sur les objets à échanges implicites IODDT de type T_COM_CHAR_BMX	110
Informations détaillées sur les objets à échanges explicites de l'IODDT de type T_COM_CHAR_BMX	111
Informations détaillées sur les objets langage associés à la configuration en mode caractère	114

## Informations détaillées sur les objets langage à échange explicite pour la communication en Mode caractère.

### Présentation

Les tableaux suivants présentent tous les objets langage de configuration pour la communication en Mode caractère. Ces objets ne sont pas intégrés dans les IODDT.

### Liste des objets à échange explicite

Le tableau ci-dessous présente les objets à échange explicite :

Adresse	Type	Accès	Signification
%MWx.m.c.4	INT	L	Anomalie dans les caractères envoyés.
%MWx.m.c.5	INT	L	Anomalie dans les caractères reçus.
%MWx.m.c.24.0	BOOL	LE	Réinitialise les compteurs d'erreur lorsque la valeur 1 est définie.
%QWx.m.c.0 = 16#DEAD	INT	LE	Redémarrer le module BMX NOM 0200.

## Informations détaillées sur les objets à échanges implicites IODDT de type T\_COM\_CHAR\_BMX

### Présentation

Les tableaux ci-dessous indiquent les objets à échanges implicites de l'IODDT de type T\_COM\_CHAR\_BMX, qui sont applicables à la communication Mode caractère.

### Bit d'erreur

Le tableau ci-dessous indique la signification du bit d'erreur CH\_ERROR (%Ir.m.c.ERR) :

Symbole standard	Type	Accès	Signification	Adresse
CH_ERROR	EBOOL	R	Bit d'erreur de voie de communication.	%Ir.m.c.ERR

### Objet de signal sur l'entrée

Le tableau ci-dessous indique la signification du bit du mot INPUT\_SIGNALS (%IW.r.m.c.0) :

Symbole standard	Type	Accès	Signification	Adresse
DCD	BOOL	L	Signal DCD RS232 (applicable seulement au module BMX NOM 0200).	%IW.r.m.c.0.0
CTS	BOOL	L	Signal RS232 CTS.	%IW.r.m.c.0.2
DSR	BOOL	L	Signal RS232 DST (applicable seulement au module BMX NOM 0200).	%IW.r.m.c.0.3

**NOTE** : %IW.r.m.c.0.2 est à 1 lorsque la tension sur le signal CTS est positive. Il s'applique également aux signaux DCD et DSR.

## Informations détaillées sur les objets à échanges explicites de l'IODDT de type T\_COM\_CHAR\_BMX

### Présentation

Cette partie présente les objets à échanges explicites de l'IODDT de type T\_COM\_CHAR\_BMX, qui sont applicables à la communication Mode caractère. Elle inclut les objets de type mot, dont les bits ont une signification particulière. Ces objets sont présentés en détail ci-dessous.

Dans cette partie, la variable IODDT\_VAR1 est du type T\_COM\_STS\_GEN.

### Observations

De manière générale, la signification des bits est donnée pour l'état 1. Dans certains cas, chaque état du bit est expliqué.

Tous les bits ne sont pas utilisés.

### Indicateur d'exécution d'échange explicite : EXCH\_STS

Le tableau ci-dessous présente la signification des bits de contrôle d'échange de la voie EXCH\_STS (MW<sub>r.m.c.0</sub>) :

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	L	Lecture des mots d'état de la voie en cours.	%MW <sub>r.m.c.0.0</sub>
CMD_IN_PROGR	BOOL	L	Echange de paramètres de commande en cours.	%MW <sub>r.m.c.0.1</sub>
ADJ_IN_PROGR	BOOL	L	Echange de paramètres de réglage en cours (non applicable au module BMX NOM 0200).	%MW <sub>r.m.c.0.2</sub>

### Compte rendu d'échange explicite : EXCH\_RPT

Le tableau ci-dessous indique la signification des bits de compte rendu d'échange EXCH\_RPT (%MW<sub>r.m.c.1</sub>) :

Symbole standard	Type	Accès	Signification	Adresse
STS_ERR	BOOL	L	Erreur détectée de lecture des mots d'état de la voie.	%MW <sub>r.m.c.1.0</sub>
CMD_ERR	BOOL	L	Anomalie lors d'un échange de paramètres de commande.	%MW <sub>r.m.c.1.1</sub>
ADJ_ERR	BOOL	L	Anomalie lors de l'échange de paramètres de réglage (non applicable au module BMX NOM 0200).	%MW <sub>r.m.c.1.2</sub>

**Défauts détectés de voie standard : CH\_FLT**

Le tableau ci-dessous présente la signification des bits du mot d'état CH\_FLT (%MWr.m.c.2) :

Symbole standard	Type	Accès	Signification	Adresse
NO_DEVICE	BOOL	L	Aucun équipement ne fonctionne sur la voie.	%MWr.m.c.2.0
ONE_DEVICE_FLT	BOOL	L	Un équipement sur la voie n'est pas fonctionnel.	%MWr.m.c.2.1
BLK	BOOL	L	Bornier non connecté.	%MWr.m.c.2.2
TO_ERR	BOOL	L	Délai d'attente dépassé (analyse nécessaire).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	L	Erreur interne détectée ou autotest de la voie.	%MWr.m.c.2.4
CONF_FLT	BOOL	L	Configurations matérielle et logicielle différentes.	%MWr.m.c.2.5
COM_FLT	BOOL	L	Analyse de la communication nécessaire sur l'automate.	%MWr.m.c.2.6
APPLI_FLT	BOOL	L	Erreur détectée d'application (erreur de réglage ou de configuration).	%MWr.m.c.2.7

La lecture est effectuée par l'instruction READ\_STS (IODDT\_VAR1).

**Etat spécifique de la voie, %MWr.m.c.3**

Le tableau ci-dessous présente la signification des bits du mot d'état de voie PROTOCOL (%MWr.m.c.3) :

Symbole standard	Type	Accès	Signification	Adresse
PROTOCOL	INT	R	Octet 0 = 16#03 pour la fonction Mode caractère.	%MWr.m.c.3

La lecture est effectuée par l'instruction READ\_STS (IODDT\_VAR1).

**Commande de voie %MWr.m.c.24**

Le tableau ci-dessous présente la signification des bits du mot CONTROL (%MWr.m.c.24) :

Symbole standard	Type	Accès	Signification	Adresse
DTR_ON	BOOL	L/E	Définit le signal DTR.	%MWr.m.c.24.8
DTR_OFF	BOOL	L/E	Réinitialise le signal DTR.	%MWr.m.c.24.9

La commande est exécutée par l'instruction WRITE\_CMD (IODDT\_VAR1).

Pour plus d'informations sur le changement de protocole, consultez la section sur les changements de protocole (*voir page 119*).



**Objet mot %QWr.m.c.0**

Le tableau ci-après présente la signification du bit 0 du mot %QWr.m.c.0 :

Symbole standard	Type	Accès	Signification	Adresse
STOP_EXCH	BOOL	L/E	Arrête tous les échanges sur front montant (disponible uniquement sur le module BMX NOM 0200).	%QWr.m.c.0.0

## Informations détaillées sur les objets langage associés à la configuration en mode caractère

### Présentation

Les tableaux suivants présentent tous les objets langage de configuration pour la communication en mode caractère. Ces objets ne sont pas intégrés aux IODDT et peuvent être affichés par le programme d'application.

### Liste des objets à échanges explicites en mode caractère

Le tableau ci-dessous répertorie les objets à échanges explicites.

Adresse	Type	Accès	Signification
%KWr.m.c.0	INT	R	L'octet 0 de ce mot correspond au type. La valeur 3 représente le mode caractère.
%KWr.m.c.1	INT	R	<p>L'octet 0 de ce mot correspond à la vitesse de transmission. Il peut avoir différentes valeurs :</p> <ul style="list-style-type: none"> <li>● La valeur -2 (0xFE) correspond à 300 bits/s.</li> <li>● La valeur -1 (0xFF) correspond à 600 bits/s.</li> <li>● La valeur 0 (0x00) correspond à 1200 bits/s.</li> <li>● La valeur 1 (0x01) correspond à 2400 bits/s.</li> <li>● La valeur 2 (0x02) correspond à 4800 bits/s.</li> <li>● La valeur 3 (0x03) correspond à 9600 bits/s (valeur par défaut).</li> <li>● La valeur 4 (0x04) correspond à 19200 bits/s.</li> <li>● La valeur 5 (0x05) correspond à 38400 bits/s.</li> <li>● La valeur 6 (0x06) correspond à 57600 bits/s (ne peut être utilisé que pour le module BMX NOM 0200)</li> <li>● La valeur 7 (0x07) correspond à 115200 bits/s (ne peut être utilisé que pour le module BMX NOM 0200)</li> </ul> <p>L'octet 1 de ce mot correspond au format :</p> <ul style="list-style-type: none"> <li>● Bit 8 : nombre de bits (1 = 8 bits (RTU), 0 = 7 bits (ASCII))</li> <li>● bit 9 = 1 : gestion de la parité (1 = avec, 0 = sans)</li> <li>● Bit 10 : type de parité (1 = impair, 0 = pair)</li> <li>● Bit 11 : nombre de bits d'arrêt (1 = 1 bit, 0 = 2 bits)</li> <li>● Bit 13 : ligne physique (1 = RS232, 0 = RS485)</li> <li>● Bit 14 : Signaux de modem DTR/DSR/DCD Pour le module BMX NOM 0200 et si RS232 est sélectionné, le bit peut avoir 2 valeurs différentes : 1 signifie que les signaux de modem sont gérés, 0 qu'ils ne le sont pas (valeur par défaut pour BMX P34 ou si RS485 est sélectionné)</li> <li>● Bit 15 : Signaux de gestion de contrôle de flux matériel RTS/CTS Si RS232 est sélectionné, 2 valeurs peuvent être associées à ce bit : 0 pour RX/TX et 1 pour RX/TX + RTS/CTS. Si RS485 est sélectionné, la valeur par défaut est 0 (RX/TX).</li> </ul>

Adresse	Type	Accès	Signification
%KWr.m.c.2	INT	R	toute valeur en ms saisie dans cette même zone, qui est fonction de la vitesse de transmission et du format sélectionnés (la valeur 0 suppose une absence de détection de silence).
%KWr.m.c.3	INT	R	Ce mot correspond au type de polarisation : <ul style="list-style-type: none"> <li>● la valeur 0 sur les deux bits 14 et 15 correspond à l'absence de polarisation (c'est une valeur par défaut pour BMX P34 ou si RS232 est sélectionné)</li> <li>● Bit 14 : la valeur 1 correspond à une polarisation à faible impédance (comparable Modbus) et ne peut être utilisée que pour le module BMX NOM 0200 et si RS485 est sélectionné</li> <li>● Bit 15 : la valeur 1 correspond à une polarisation à haute impédance et ne peut être utilisée que pour le module BMX NOM 0200 et si RS485 est sélectionné</li> </ul>
%KWr.m.c.5	INT	R	Ce mot correspond, si RS232 est sélectionné, au retard RTS/CTS exprimé en millisecondes (entre 0 et 100). Si RS485 est sélectionné, la valeur par défaut est 0.
%KWr.m.c.6	INT	R	Deux valeurs peuvent être associées au bit 0 de l'octet 0 : <ul style="list-style-type: none"> <li>● 1, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 1 est activée</li> <li>● 0, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 1 est désactivée</li> </ul> Deux valeurs peuvent être associées au bit 1 de l'octet 0 : <ul style="list-style-type: none"> <li>● 1, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 1 est activée</li> <li>● 0, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 1 est désactivée</li> </ul> L'octet 1 de ce mot correspond à la valeur entrée (comprise entre 0 et 255) pour l'arrêt en réception pour le caractère 1.
%KWr.m.c.7	INT	R	Deux valeurs peuvent être associées au bit 0 de l'octet 0 : <ul style="list-style-type: none"> <li>● 1, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 2 est activée</li> <li>● 0, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 2 est désactivée</li> </ul> Deux valeurs peuvent être associées au bit 1 de l'octet 0 : <ul style="list-style-type: none"> <li>● 1, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 2 est activée</li> <li>● 0, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 2 est désactivée</li> </ul> L'octet 1 de ce mot correspond à la valeur entrée (comprise entre 0 et 255) pour l'arrêt en réception pour le caractère 2.

## Sous-chapitre 6.5

### Type d'IODDT Type T\_GEN\_MOD applicable à tous les modules

#### Informations détaillées sur les objets langage de l'IODDT de type T\_GEN\_MOD

##### Présentation

Les modules Modicon X80 sont associés à un IODDT de type T\_GEN\_MOD.

##### Observations

De manière générale, la signification des bits est donnée pour l'état 1 de ce bit. Dans les cas spécifiques, chaque état du bit est expliqué.

Certains bits ne sont pas utilisés.

##### Liste d'objets

Le tableau ci-dessous présente les différents objets de l'IODDT.

Symbole standard	Type	Accès	Signification	Adresse
MOD_ERROR	BOOL	L	Bit erreur détectée module	%I.r.m.MOD.ERR
EXCH_STS	INT	R	Mot de commande d'échange de module	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	L	Lecture des mots d'état du module en cours	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Mot de compte rendu de l'échange	%MWr.m.MOD.1
STS_ERR	BOOL	L	Evénement lors de la lecture des mots d'état du module	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Mot d'erreurs internes détectées du module	%MWr.m.MOD.2
MOD_FAIL	BOOL	L	module inutilisable	%MWr.m.MOD.2.0
CH_FLT	BOOL	L	Voie(s) inutilisable(s)	%MWr.m.MOD.2.1
BLK	BOOL	L	Bornier incorrectement câblé	%MWr.m.MOD.2.2
CONF_FLT	BOOL	L	Anomalie de configuration matérielle ou logicielle	%MWr.m.MOD.2.5
NO_MOD	BOOL	L	Module absent ou inopérant	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	L	Mot d'erreurs internes détectées du module (extension Fipio uniquement)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	L	Erreur interne détectée, module hors service (extension Fipio uniquement)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	L	Voie(s) inutilisable(s) (extension Fipio uniquement)	%MWr.m.MOD.2.9

---

<b>Symbole standard</b>	<b>Type</b>	<b>Accès</b>	<b>Signification</b>	<b>Adresse</b>
BLK_EXT	BOOL	L	Bornier incorrectement câblé (extension Fipio uniquement)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	L	Anomalie de configuration matérielle ou logicielle (extension Fipio uniquement)	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	L	Module manquant ou hors service (extension Fipio uniquement)	%MWr.m.MOD.2.14



---

# Chapitre 7

## Changement dynamique de protocoles

---

### Changement de protocole avec les processeurs Modicon M340

#### Général

Cette partie décrit comment changer le protocole utilisé par une communication série d'UC à l'aide de la commande `WRITE_CMD (IODDT_VAR1)`. Cette commande permet de basculer entre les trois protocoles suivants :

- Twido
- Maître Modbus
- Mode caractère

**NOTE** : La variable `IODDT_VAR1` doit être de type `T_COM_MB_BMX`.

#### Principe du changement de protocole

**NOTE** : Pour changer de protocole, le protocole doit initialement être configuré en mode Esclave Modbus.

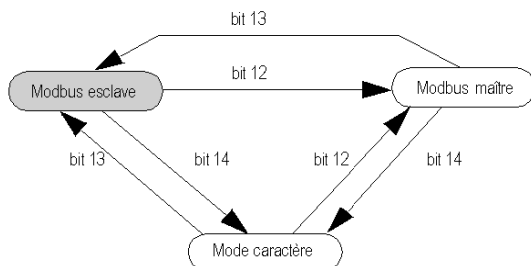
Vous devez d'abord créer une variable d'IODDT liée à la voie série du processeur, puis mettre à 1 le bit du mot `IODDT_VAR1.CONTROL (%MWr.m.c.24)` qui correspond au changement de protocole souhaité :

- `TO_MODBUS_MASTER` (bit 12) : le protocole actuel est remplacé par Maître Modbus.
- `TO_MODBUS_SLAVE` (bit 13) : le protocole actuel est remplacé par Esclave Modbus.
- `TO_CHAR_MODE` (bit 14) : le protocole actuel est remplacé par Mode caractère.

**NOTE** : `IODDT_VAR1.CONTROL (%MWr.m.c.24)` fait partie de la variable `IODDT_VAR1`.

Ensuite, appliquez l'instruction `WRITE_CMD` à la variable d'IODDT liée à la voie série du processeur.

Le schéma ci-dessous montre le changement de protocole à effectuer en fonction des bits du mot IODDT\_VAR1.CONTROL (§MWR.m.c.24) mis à 1 :



## Utilisations

Trois changements de protocole sont utilisés :

- Transfert vers le protocole Maître Modbus : le changement de protocole s'effectue en deux étapes :
  - Transfert de la configuration Esclave Modbus vers la configuration Maître Modbus
  - Retour à la configuration Esclave Modbus initiale

L'objectif de la configuration Maître Modbus est d'envoyer des informations sur un événement à un autre automate. Lorsque la configuration Esclave Modbus est remplacée par la configuration Maître Modbus, les paramètres de transmission, de signaux et de ligne physique restent les mêmes. Seules les valeurs des paramètres suivants, propres à la configuration Maître Modbus, sont modifiées :

- Le délai entre les trames reprend sa valeur par défaut, qui dépend de la vitesse de transmission.
- Le délai de réponse est réglé sur 3 000 ms.
- Le nombre de nouvelles tentatives est réglé sur 3.
- Transfert vers le mode caractère : ce changement de protocole s'effectue en deux étapes :
  - Transfert de la configuration Esclave Modbus vers la configuration Mode caractère
  - Retour à la configuration Esclave Modbus initiale

L'objectif de la configuration Mode caractère est de communiquer avec un protocole privé (un modem, par exemple). Lorsque la configuration Esclave Modbus est remplacée par la configuration Mode caractère, les paramètres de transmission, de signaux et de ligne physique restent les mêmes. Seul le paramètre de fin de message, propre au Mode caractère, prend la valeur Arrêt sur silence avec un délai d'expiration de 1 000 ms.



- Transfert vers les protocoles Mode caractère et Maître Modbus : ce changement de protocole s'effectue en trois étapes :
  - Transfert de la configuration Esclave Modbus vers la configuration Mode caractère.
  - Transfert de la configuration Mode caractère vers la configuration Maître Modbus.
  - Retour à la configuration Esclave Modbus initiale

L'objectif de la configuration Mode caractère est de communiquer avec un protocole privé (un modem, par exemple). Une fois l'échange terminé, l'utilisateur bascule vers la configuration Maître Modbus pour envoyer des informations sur un événement à un autre automate. Une fois le message envoyé, l'utilisateur revient à la configuration Esclave Modbus initiale.

**NOTE** : dans ces trois cas, la configuration par défaut reste Esclave Modbus.

### Redémarrages à froid et à chaud

Les changements de protocole ne sont pas affectés par les bits %S0 et %S1 (mis à 1 pendant un redémarrage à froid et à chaud respectivement). Cependant, un démarrage à froid ou à chaud de l'automate va configurer le port série avec ses valeurs par défaut ou avec les valeurs programmées dans l'application.



---

## Partie III

### Mise en route : Exemple de mise en œuvre d'une liaison série

---

#### Présentation

Cette partie présente un exemple de mise en œuvre d'une liaison série.

#### Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
8	Description de l'application	125
9	Installation de l'application avec Control Expert	127
10	Démarrage de l'application	155



---

# Chapitre 8

## Description de l'application

---

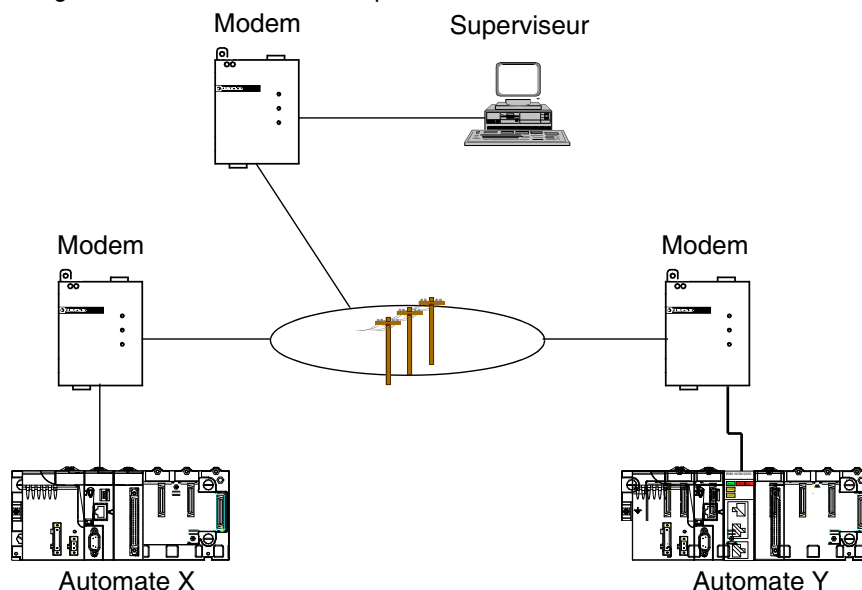
### Vue d'ensemble de l'application

#### Vue d'ensemble

L'application décrite dans ce document est une application de communication Modbus via des modems.

#### Exemple

La figure ci-dessous illustre l'exemple :



Les périphériques communiquent les uns avec les autres à l'aide de modems. Le superviseur est le Modbus maître et les automates X et Y sont les esclaves.

L'objectif de l'exemple est d'écrire les valeurs des zones de données de l'automate X vers l'automate Y.

Pour cela, l'automate X doit devenir le Modbus maître.

Le superviseur communique quotidiennement avec les automates pour récupérer des informations.

Si une alarme est déclenchée sur l'automate X, il passe en mode maître Modbus et envoie des données à l'automate Y.

Pour simplifier la programmation, les modems ont été initialisés à l'aide des paramètres adéquats par l'intermédiaire d'un terminal de programmation. Les paramètres sont sauvegardés dans la mémoire non volatile à l'aide des commandes AT et W.

### Mode de fonctionnement

Le fonctionnement de l'application est le suivant :

Etape	Action
1	Le port de l'automate X est défini en mode caractère.
2	L'automate X envoie un message de composition au modem.
3	Le port de l'automate X est défini en mode maître Modbus.
4	L'automate maître (X) envoie des données à l'automate esclave (Y).
5	Le port est défini en mode caractère.
6	L'automate X envoie un message de déconnexion au modem.
7	Le port de l'automate X est défini en mode esclave Modbus.

---

# Chapitre 9

## Installation de l'application avec Control Expert

---

### Objet du chapitre

Ce chapitre décrit la procédure à suivre pour créer l'application décrite. Les étapes à suivre pour créer les différents éléments de l'application sont présentées de façon générale, mais également de façon détaillée.

### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
9.1	Présentation de la solution utilisée	128
9.2	Développement de l'application	129

# Sous-chapitre 9.1

## Présentation de la solution utilisée

---

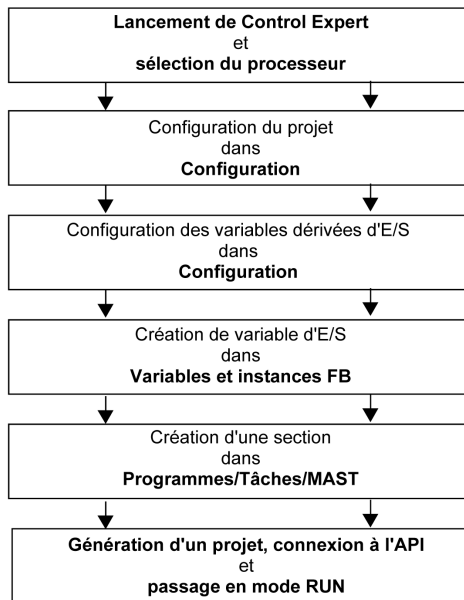
### Différentes étapes du processus utilisant Control Expert

#### Présentation

Le logigramme ci-dessous présente les différentes étapes à suivre pour créer l'application. Vous devez respecter un ordre chronologique afin de définir correctement tous les éléments de l'application.

#### Description

Description des différents types :





---

## Sous-chapitre 9.2

### Développement de l'application

---

#### Objet de cette section

Cette section décrit pas à pas la création de l'application à l'aide de Control Expert.

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Création du projet	130
Déclaration des variables	135
Utilisation d'un modem	139
Procédure de programmation	141
Structure de programmation	143
Programmation	146

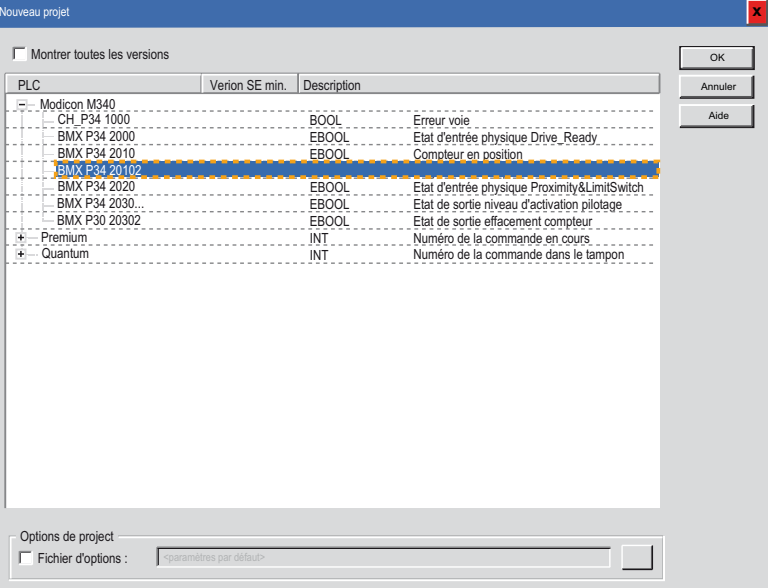
## Création du projet

### Présentation

Pour poursuivre le développement de l'exemple, il faut créer un projet principal associé à l'automate X pour configurer l'automate X, ainsi que pour déclarer toutes les variables nécessaires et programmer l'application. De plus, il faut créer un projet séparé pour la configuration de l'automate Y.

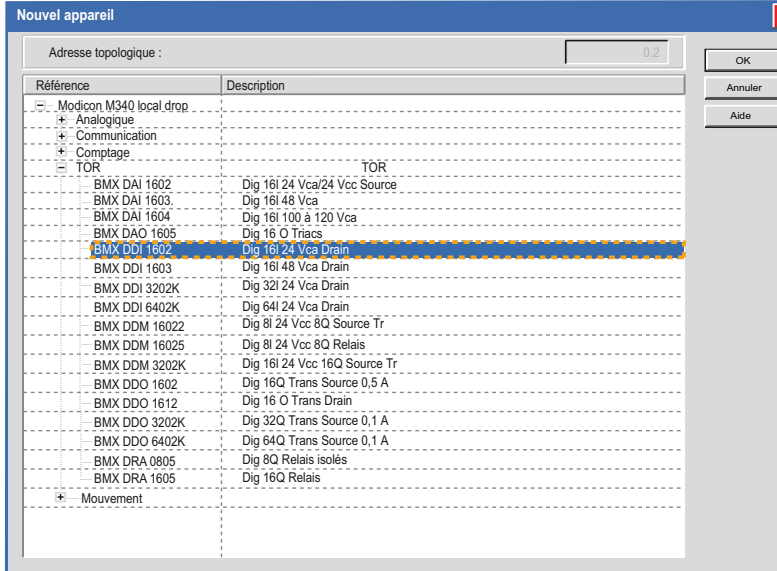
### Marche à suivre pour créer un projet

Le tableau ci-dessous présente la procédure pour créer un projet à l'aide de Control Expert.

Etape	Action																																	
1	Lancez le logiciel Control Expert.																																	
2	<p>Cliquez sur Fichier, puis sur Nouveau et choisissez un processeur BMX P34 20102 :</p>  <table border="1" data-bbox="312 698 971 889"> <thead> <tr> <th>PLC</th> <th>Version SE min.</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Modicon M340</td> <td></td> <td></td> </tr> <tr> <td>CH P34 1000</td> <td></td> <td>BOOL Erreur voie</td> </tr> <tr> <td>BMX P34 2000</td> <td></td> <td>EBOOL Etat d'entrée physique Drive_Ready</td> </tr> <tr> <td>BMX P34 2010</td> <td></td> <td>EBOOL Compteur en position</td> </tr> <tr style="border: 2px dashed blue;"> <td><b>BMX P34 20102</b></td> <td></td> <td></td> </tr> <tr> <td>BMX P34 2020</td> <td></td> <td>EBOOL Etat d'entrée physique Proximity&amp;LimitSwitch</td> </tr> <tr> <td>BMX P34 2030...</td> <td></td> <td>EBOOL Etat de sortie niveau d'activation pilotage</td> </tr> <tr> <td>BMX P30 20302</td> <td></td> <td>EBOOL Etat de sortie effacement compteur</td> </tr> <tr> <td>Premium</td> <td></td> <td>INT Numéro de la commande en cours</td> </tr> <tr> <td>Quantum</td> <td></td> <td>INT Numéro de la commande dans le tampon</td> </tr> </tbody> </table> <p>Options de projet  <input type="checkbox"/> Fichier d'options : <input type="text" value="paramètres par défaut"/></p>	PLC	Version SE min.	Description	Modicon M340			CH P34 1000		BOOL Erreur voie	BMX P34 2000		EBOOL Etat d'entrée physique Drive_Ready	BMX P34 2010		EBOOL Compteur en position	<b>BMX P34 20102</b>			BMX P34 2020		EBOOL Etat d'entrée physique Proximity&LimitSwitch	BMX P34 2030...		EBOOL Etat de sortie niveau d'activation pilotage	BMX P30 20302		EBOOL Etat de sortie effacement compteur	Premium		INT Numéro de la commande en cours	Quantum		INT Numéro de la commande dans le tampon
PLC	Version SE min.	Description																																
Modicon M340																																		
CH P34 1000		BOOL Erreur voie																																
BMX P34 2000		EBOOL Etat d'entrée physique Drive_Ready																																
BMX P34 2010		EBOOL Compteur en position																																
<b>BMX P34 20102</b>																																		
BMX P34 2020		EBOOL Etat d'entrée physique Proximity&LimitSwitch																																
BMX P34 2030...		EBOOL Etat de sortie niveau d'activation pilotage																																
BMX P30 20302		EBOOL Etat de sortie effacement compteur																																
Premium		INT Numéro de la commande en cours																																
Quantum		INT Numéro de la commande dans le tampon																																
3	Validez par OK.																																	

### Sélection de module d'entrée TOR

Le tableau ci-dessous présente la procédure de sélection du module TOR dont a besoin l'automate X.

Etape	Action
1	Dans le Navigateur du projet, faites un double-clic sur Configuration, puis sur 0:Bus automate et sur 0:BMX XBP ... (0 correspondant au numéro du rack).
2	Dans la fenêtre Bus automate, sélectionnez un emplacement (par exemple, l'emplacement 1) et cliquez deux fois dessus.
3	<p>Choisissez le module d'entrée TOR BMX DDI 1602 situé dans la liste de modules TOR.</p> 
4	Validez par OK.

### Sélection de module BMX NOM 0200

Dans cet exemple, un module BMX NOM 0200 est utilisé dans l'automate Y pour la liaison série avec le modem. Il faut donc l'ajouter au projet associé à l'automate Y.

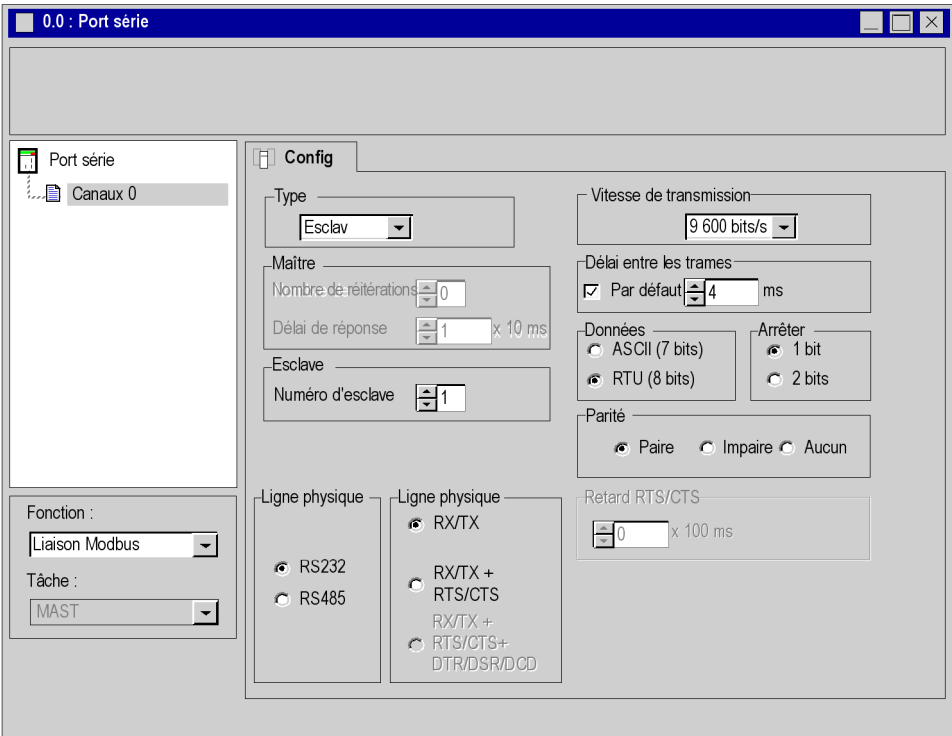
Le tableau ci-dessous présente la procédure à suivre pour sélectionner le module BMX NOM 0200.

Etape	Action																										
1	Dans le Navigateur du projet, faites un double-clic sur Configuration, puis sur 0:Bus automate et sur 0:BMX XBP ... (0 correspondant au numéro du rack).																										
2	Dans la fenêtre Bus automate, sélectionnez un emplacement (par exemple, l'emplacement 1) et cliquez deux fois dessus.																										
3	<p>Choisissez le module de communication BMX NOM 0200 situé dans la liste de modules Communication.</p> <thead> <tr> <th>Référence de produit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Modicon M340 local drop</td> <td></td> </tr> <tr> <td>  + Analogique</td> <td></td> </tr> <tr> <td>  - Communication</td> <td></td> </tr> <tr> <td>    BMX EIA 100</td> <td>Module d'interface AS V3</td> </tr> <tr> <td>    BMX NOE 0100</td> <td>Port 10/100RJ45 Ethernet1</td> </tr> <tr> <td>    BMX NOE 0100.2</td> <td>Port 10/100RJ45 Ethernet1</td> </tr> <tr> <td>    BMX NOE 0110</td> <td>Port 10/100RJ45 Ethernet1</td> </tr> <tr> <td>    BMX NOE 0110.2</td> <td>Port 10/100RJ45 Ethernet1</td> </tr> <tr> <td>    BMX NOM 0200</td> <td>Module de bus 2 ports RS485/232</td> </tr> <tr> <td>  + Comptage</td> <td></td> </tr> <tr> <td>  + TOR</td> <td></td> </tr> <tr> <td>  + Mouvement</td> <td></td> </tr> </tbody>	Référence de produit	Description	Modicon M340 local drop		+ Analogique		- Communication		BMX EIA 100	Module d'interface AS V3	BMX NOE 0100	Port 10/100RJ45 Ethernet1	BMX NOE 0100.2	Port 10/100RJ45 Ethernet1	BMX NOE 0110	Port 10/100RJ45 Ethernet1	BMX NOE 0110.2	Port 10/100RJ45 Ethernet1	BMX NOM 0200	Module de bus 2 ports RS485/232	+ Comptage		+ TOR		+ Mouvement	
Référence de produit	Description																										
Modicon M340 local drop																											
+ Analogique																											
- Communication																											
BMX EIA 100	Module d'interface AS V3																										
BMX NOE 0100	Port 10/100RJ45 Ethernet1																										
BMX NOE 0100.2	Port 10/100RJ45 Ethernet1																										
BMX NOE 0110	Port 10/100RJ45 Ethernet1																										
BMX NOE 0110.2	Port 10/100RJ45 Ethernet1																										
BMX NOM 0200	Module de bus 2 ports RS485/232																										
+ Comptage																											
+ TOR																											
+ Mouvement																											

| 4 | Validez par OK. |

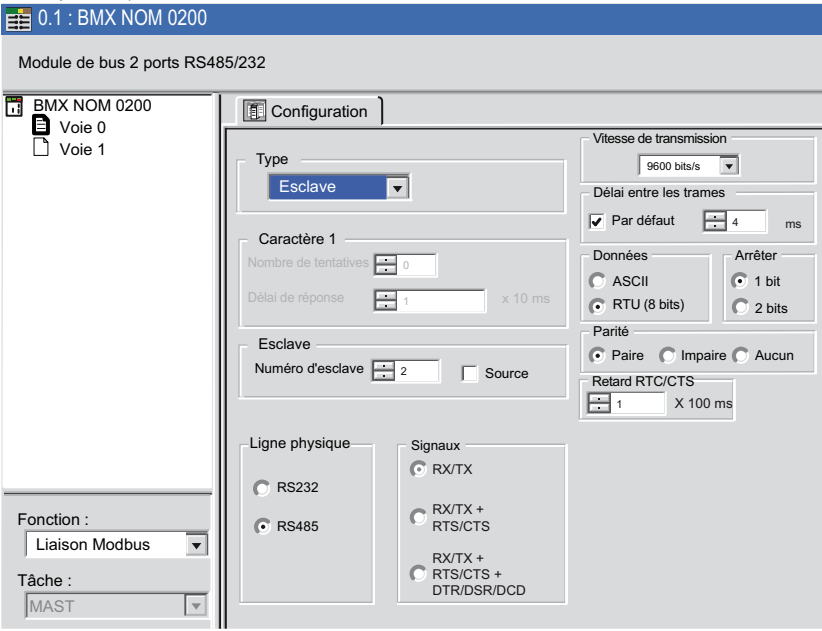
### Configuration du port série du processeur

Le tableau ci-dessous présente la procédure de configuration du port série du processeur d'automate X en tant qu'esclave Modbus :

Etape	Action
1	<p>Dans le Navigateur de projet, double-cliquez sur Configuration, sur 0:BMX XBP 0800 et sur 0:BMX P34 20102. Puis double-cliquez sur Port série pour accéder à la fenêtre 0.0:Port série.</p> 
2	Sélectionnez le type <b>Esclave</b> .
3	Sélectionnez le débit de transmission <b>9600 bits/s</b> .
4	Sélectionnez la ligne physique <b>RS232</b> .
5	Sélectionnez le type de données <b>RTU (8 bits)</b> .
6	Fermez la fenêtre et validez en cliquant sur <b>OK</b> .

### Configuration de la voie série BMX NOM 0200

Le tableau ci-dessous présente la procédure de configuration de la voie série du module BMX NOM 0200 de l'automate Y en tant qu'esclave Modbus :

Etape	Action
1	<p>Dans le Navigateur de projet, double-cliquez sur Configuration, sur 0:BMX XBP 0800 puis sur 0:BMX NOM 0200 pour accéder à la fenêtre 0.x:BMX NOM 0200 (x étant le numéro de l'emplacement, par exemple x=1).</p> 
2	Sélectionnez la Voie 0.
3	Sélectionnez la fonction Liaison Modbus.
4	Sélectionnez le type Esclave.
5	Sélectionnez le débit de transmission 9600 bits/s.
6	Sélectionnez la ligne physique RS232.
7	Sélectionnez RX/TX + RTS/CTS + DTR/DSR/DCD pour les signaux.
8	Sélectionnez 100 ms comme temporisation RTS/CTS.
9	Sélectionnez le type de données RTU (8 bits).
10	Fermez la fenêtre et validez en cliquant sur OK.

## Déclaration des variables

### Présentation

Toutes les variables utilisées dans les différentes sections du programme doivent être déclarées. Les variables non déclarées ne peuvent pas être utilisées dans le programme.

**NOTE :** Pour plus d'informations, consultez le chapitre *Editeur de données (voir EcoStruxure™ Control Expert, Modes de fonctionnement)*.

### Procédure de déclaration des variables

Le tableau ci-dessous présente la procédure à suivre pour déclarer des variables d'application :

Etape	Action
1	Dans <i>Navigateur de projet/Variables et instances FB</i> , double-cliquez sur <i>Variables élémentaires</i> .
2	Dans la fenêtre <i>Editeur de données</i> , sélectionnez la case dans la colonne <i>Nom</i> , puis entrez le nom de votre première variable.
3	Sélectionnez à présent un type de variable.
4	Une fois toutes vos variables déclarées, vous pouvez fermer la fenêtre.

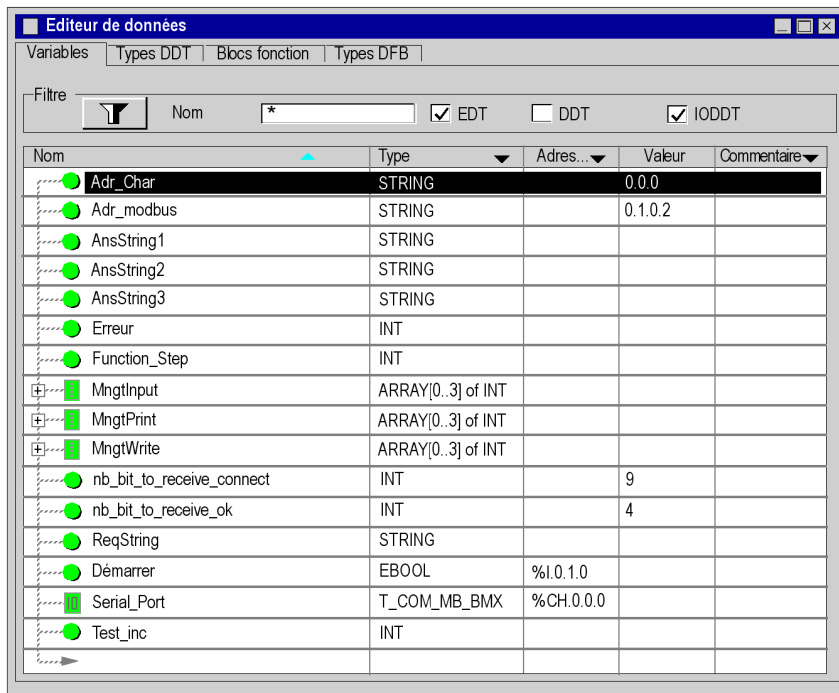
### Variables utilisées pour l'application

Le tableau ci-dessous présente les détails de variables utilisées dans l'application et déclarées dans le projet associé à l'automate X :

Variable	Type	Définition
Adr_Char	STRING	Adresse du port série de l'automate maître
Adr_modbus	STRING	Adresse de voie série de l'automate esclave Modbus (voie 0 du module BMX NOM 0200).
AnsString1	STRING	Premier caractère de réponse du modem
AnsString2	STRING	Deuxième caractère de réponse du modem
AnsString3	STRING	Troisième caractère de réponse du modem
Erreur	INT	Code d'erreur fonction
Function_Step	INT	Echelon
MngtInput	ARRAY[0..3] of INT	Tableau des paramètres de communication du bloc INPUT_CHAR
MngtPrint	ARRAY[0..3] of INT	Tableau des paramètres de communication du bloc PRINT_CHAR
MngtWrite	ARRAY[0..3] of INT	Tableau des paramètres de communication du bloc WRITE_VAR

Variable	Type	Définition
nb_charac_to_receive_connect	INT	Nombre de caractères à recevoir : connexion au modem
nb_charac_to_receive_ok	INT	Nombre de caractères à recevoir : message de confirmation du modem
ReqString	STRING	Réponse du modem
Start	EBOOL	Mode de démarrage (signal provenant de la voie 0 du module BMX DDI 1602).
Serial_Port	T_COM_MB_BMX	Objet d'E/S du port série
Test_inc	INT	Valeur d'incréméntation

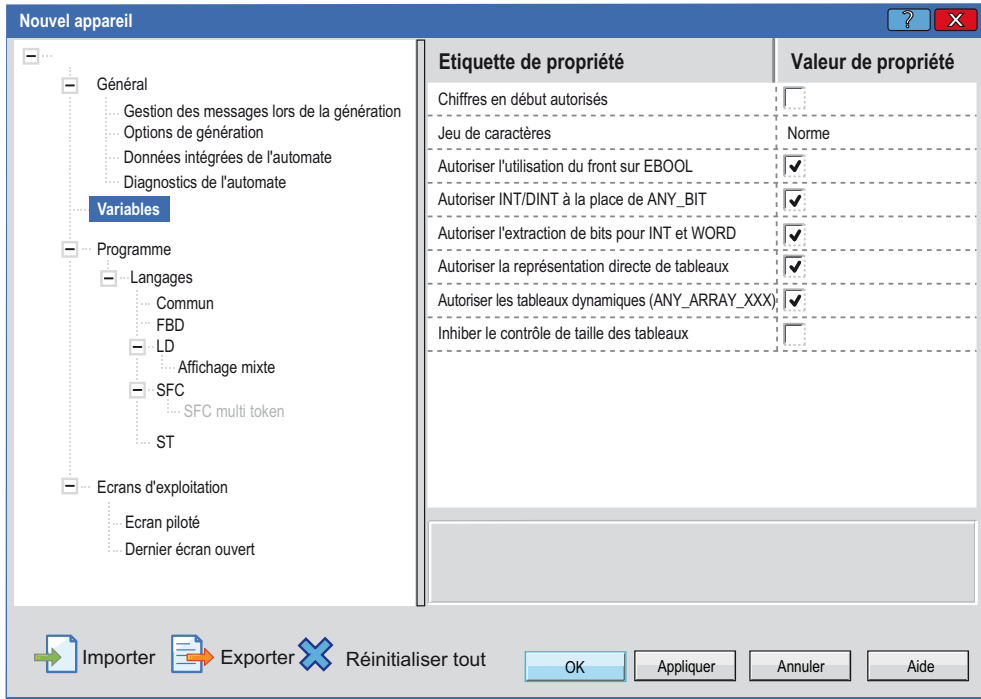
L'écran ci-dessous présente les variables d'application créées à l'aide de l'éditeur de données :




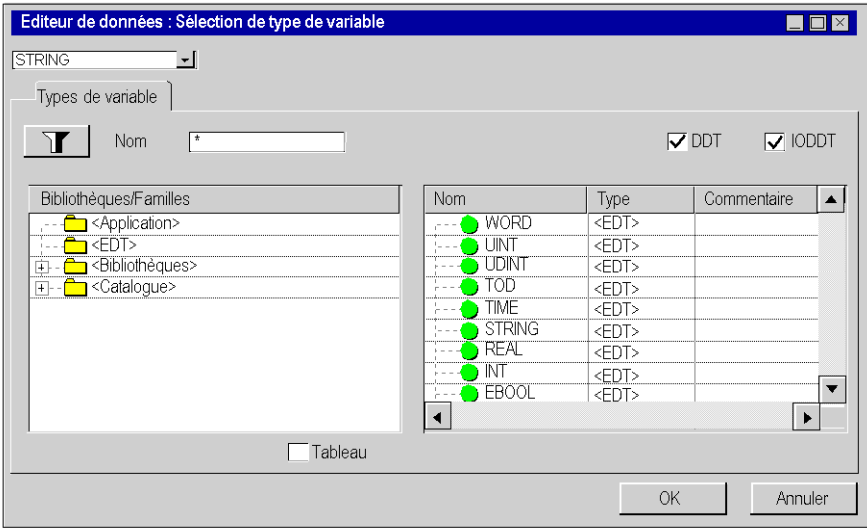



## Déclaration d'un type de tableau

Avant de déclarer un type de tableau, cliquez sur **Outils/Options du projet/Variables**, puis sélectionnez les options "Autoriser la représentation directe de tableaux" et "Autoriser les tableaux dynamiques".



Le tableau suivant présente la manière de déclarer le type de tableau :

Etape	Action
1	Dans le Navigateur de projet, cliquez sur Variables et instances FB.
2	Cliquez dans la colonne Nom et entrez le nom de la variable.
3	<p>Double-cliquez dans la colonne Type, puis cliquez sur le bouton .</p> <p>La fenêtre Sélection de type de variable s'ouvre :</p> 
4	<p>Choisissez le type de variable souhaité (par exemple, cliquez sur &lt;EDT&gt; et sélectionnez INT), puis cliquez sur la case à cocher Tableau.</p> 
5	Modifiez l'intervalle, puis confirmez-le par OK.

### Déclaration des objets d'E/S

Pour déclarer des variables dérivées d'E/S, ouvrez la fenêtre Sélection de type de variable comme indiqué dans la procédure ci-dessus et cliquez sur <Catalogue> pour accéder aux variables de type <IODDT> (sélectionnez par exemple T COM MB BMX), puis confirmez par OK.

## Utilisation d'un modem

### Description

Trois commandes sont nécessaires à l'interfaçage des modems téléphoniques sur les automates. Ces commandes sont les suivantes :

- initialisation du modem
- renumérotation
- déconnexion du modem

Il est impératif d'envoyer un message d'initialisation suivi d'un message de composition au modem avant de lui envoyer un message ASCII ou Modbus.

Une fois la connexion établie entre les deux modems, vous pouvez envoyer un nombre illimité de messages ASCII ou Modbus.

Lorsque tous les messages ont été envoyés, vous devez envoyer la chaîne de déconnexion au modem.

### Initialisation du modem

Les deux modems doivent être configurés avec les mêmes caractéristiques que les ports série :

- vitesse de données : 9600 bauds,
- bloc de caractère : 8 bits / parité paire / 1 bit d'arrêt
- modulation de ligne : V32

Définissez ensuite "+" en tant que caractère d'échappement (commande : ATS2=43).

Exemple de commande d'initialisation :

```
ATQ0&Q0E0&K0V1
```

Avec :

- Q0 : activer le code de résultat
- &Q0 : DTR est toujours considéré (ACTIVE)
- E0 : désactiver l'écho de caractères
- &K0 : pas de contrôle de flux
- V1 : codes de résultat sous forme de mots

### Numérotation du modem

Le message de composition est utilisé pour envoyer le numéro de téléphone au modem.

Seules les commandes AT liées à la composition d'un numéro doivent être incluses dans le message.

Exemple :

- **Numérotation par fréquence** : ATDT6800326<CR><LR>
- **Numérotation par impulsion** : ATDP6800326<CR><LF>
- **Numérotation par fréquence avec attente de tonalité** : ATDTW6800326<CR><LF>

### Déconnexion du modem

Le modem est d'abord remis en mode commande en recevant trois fois le caractère d'échappement.

La commande de déconnexion "ATH0" peut ensuite être envoyée.

**Séquence d'échappement :** "+++" (code de résultat du modem : OK),

**Commande de déconnexion :** "ATH0" (code de résultat du modem : OK).

## Procédure de programmation

### Procédure

Le tableau ci-dessous présente la procédure de programmation de l'application.

Etape	Action	Détails
1	Préparation du port de communication.	<ul style="list-style-type: none"> <li>● Définissez le Modbus esclave en mode caractère en envoyant WRITE_CMD (<i>voir page 142</i>) au port série.</li> <li>● Pour une transmission par modem, envoyez la commande HAYES en utilisant le bloc PRINT_CHAR pour configurer le modem (<i>voir page 139</i>).</li> <li>● Pour une transmission par modem, envoyez la commande HAYES en utilisant le bloc PRINT_CHAR. Le message de composition est utilisé pour envoyer le numéro de téléphone au modem (<i>voir page 139</i>).</li> </ul>
2	Mode Modbus maître	<ul style="list-style-type: none"> <li>● Passez en mode Modbus maître à l'aide de la fonction WRITE_CMD (<i>voir page 142</i>).</li> <li>● Envoyez les données à écrire sur l'automate esclave.</li> </ul>
3	Réinitialisation du port de communication.	<ul style="list-style-type: none"> <li>● Passez en mode caractère à l'aide de la fonction WRITE_CMD (<i>voir page 142</i>).</li> <li>● Pour une transmission par modem, envoyez le caractère d'échappement, puis la commande de déconnexion pour envoyer un message de déconnexion au modem (<i>voir page 140</i>) en utilisant le bloc PRINT_CHAR.</li> <li>● Revenez au mode initial du port série (Modbus esclave) à l'aide de la commande WRITE_CMD (<i>voir page 142</i>).</li> </ul>

### Écriture des mots de commande

Les étapes suivantes permettent d'envoyer une commande WRITE\_CMD vers un port de communication :

Étape	Action	Détail
1	Test permettant de déterminer si une commande est en attente.	Avant d'exécuter la commande WRITE_CMD, effectuez le test permettant de déterminer si un échange est en cours à l'aide de l'objet langage EXCH_STS (%MWr.m.c.0). Pour actualiser le mot, utilisez le bloc READ_STS.
2	Affectation du mot de commande.	Vous devez ensuite modifier la valeur de l'objet langage de commande afin d'exécuter la commande requise. Pour une liaison Modbus, l'objet langage correspond au mot interne CONTROL (%MWr.m.c.24). Par exemple, pour passer du mode Modbus au mode caractère, le bit 14 du mot %MWr.m.c.24 est défini sur 1. <b>Remarque :</b> Un bit de commande unique doit ensuite être défini de 0 à 1 avant de transmettre la commande WRITE_CMD.
3	Envoi de la commande	Enfin, WRITE_CMD doit être exécuté pour acquitter la commande.

## Structure de programmation

### Commentaires sur les étapes

Numéro de l'étape	Description de l'étape	Élément
0	Etat initial de fonction Quand le bit de démarrage passe à 1, initialiser l'erreur à 0 et passer à l'étape 5.	Modem
5	Lire l'état du port série et vérifier qu'aucune commande n'est active. Passer en mode caractère et initialiser le compteur Test_inc à 0. Allez à l'étape 10.	
10	Lire l'état du port série et vérifier qu'aucune commande n'est active. Réinitialiser le bit de commande TO_CHAR_MODE. <ul style="list-style-type: none"> <li>● S'il n'y a pas d'erreur sur le port série <ul style="list-style-type: none"> <li>○ et si le mode caractère est actif, passer à l'étape 15.</li> <li>○ et le mode caractère n'est pas actif, incrémenter Test_inc et recommencer l'étape 10 jusqu'à 1000 fois. Après 1000 échecs de tentatives, affecter Erreur à 10 et passer à l'étape 130.</li> </ul> </li> <li>● S'il y a une erreur sur le port série, alors <ul style="list-style-type: none"> <li>○ affecter Erreur à 10.</li> <li>○ Allez à l'étape 130.</li> </ul> </li> </ul>	
15	Envoyer une commande de numérotation au modem par le bloc PRINT_CHAR. Allez à l'étape 20.	
20	Si le résultat de PRINT_CHAR est concluant, passer à l'étape 25 ; sinon, définir Erreur à 20 et passer à l'étape 130.	
25	Attente de la réponse du modem par le bloc INPUT_CHAR. Quand la chaîne de réponse est totalement reçue, passer à l'étape 30.	
30	Si le résultat de INPUT_CHAR est concluant, passer à l'étape 35 ; sinon, définir Erreur à 30 et passer à l'étape 130.	
35	Si le modem répond correctement, passer à l'étape 40 ; sinon, définir Erreur à 35 et passer à l'étape 130.	

Numéro de l'étape	Description de l'étape	Elément
40	Lire l'état du port série et vérifier qu'aucune commande n'est active. Passer au mode maître Modbus et initialisez le compteur Test_inc à 0. Allez à l'étape 45.	Mode maître Modbus
45	Lire l'état du port série et vérifier qu'aucune commande n'est active. Réinitialiser le bit de commande TO_CHAR_MODE. <ul style="list-style-type: none"> <li>● S'il n'y a pas d'erreur sur le port série <ul style="list-style-type: none"> <li>○ et si le mode caractère est actif, passer à l'étape 50.</li> <li>○ et le mode caractère n'est pas actif, incrémenter Test_inc et recommencer l'étape 45 jusqu'à 1000 fois. Après 1000 échecs de tentatives, affecter Erreur à 45 et passer à l'étape 130.</li> </ul> </li> <li>● S'il y a une erreur sur le port série, alors <ul style="list-style-type: none"> <li>○ affecter Erreur à 45.</li> <li>○ Allez à l'étape 130.</li> </ul> </li> </ul>	
50	Initialisation du paramètre du bloc WRITE_VAR. Envoyer les données à écrire sur l'automate à l'aide de la fonction WRITE_VAR. Allez à l'étape 55.	Fonction d'écriture
55	Si le résultat de WRITE_VAR est concluant, passer à l'étape 60 ; sinon, définir Erreur à 55 et passer à l'étape 130.	
60	Lire l'état du port série et vérifier qu'aucune commande n'est active. Passer en mode caractère et initialiser le compteur Test_inc à 0. Allez à l'étape 65	Mode caractère
65	Lire l'état du port série et vérifier qu'aucune commande n'est active. Réinitialiser le bit de commande TO_CHAR_MODE. <ul style="list-style-type: none"> <li>● S'il n'y a pas d'erreur sur le port série <ul style="list-style-type: none"> <li>○ et si le mode caractère est actif, passer à l'étape 70.</li> <li>○ et le mode caractère n'est pas actif, incrémenter Test_inc et recommencer l'étape 65 jusqu'à 1000 fois. Après 1000 échecs de tentatives, affecter Erreur à 65 et passer à l'étape 130.</li> </ul> </li> <li>● S'il y a une erreur sur le port série, alors <ul style="list-style-type: none"> <li>○ affecter Erreur à 65.</li> <li>○ Allez à l'étape 130.</li> </ul> </li> </ul>	



Numéro de l'étape	Description de l'étape	Élément
70	Envoyer une séquence d'échappement au modem à l'aide du bloc PRINT_CHAR. Allez à l'étape 75.	Modem
75	Si le résultat de PRINT_CHAR est concluant, passer à l'étape 80 ; sinon, définir Erreur à 75 et passer à l'étape 130.	
80	Attente de la réponse du modem par le bloc INPUT_CHAR. Quand la chaîne de réponse est totalement reçue, passer à l'étape 85.	
85	Si le résultat de INPUT_CHAR est concluant, passer à l'étape 90 ; sinon, définir Erreur à 85 et passer à l'étape 130.	
90	Si le modem répond correctement, passer à l'étape 95 ; sinon, définir Erreur à 90 et passer à l'étape 130.	
95	Envoyer une commande de déconnexion au modem à l'aide du bloc PRINT_CHAR. Allez à l'étape 100.	
100	Si le résultat de PRINT_CHAR est concluant, passer à l'étape 105 ; sinon, définir Erreur à 100 et passer à l'étape 130.	
105	Attente de la réponse du modem par le bloc INPUT_CHAR. Quand la chaîne de réponse est totalement reçue, passer à l'étape 110.	
110	Si le résultat de INPUT_CHAR est concluant, passer à l'étape 115 ; sinon, définir Erreur à 110 et passer à l'étape 130.	
115	Si le modem répond correctement, passer à l'étape 120 ; sinon, définir Erreur à 115 et passer à l'étape 130.	
120	Lire l'état du port série et vérifier qu'aucune commande n'est active. Passer au mode esclave Modbus et initialiser le compteur Test_inc à 0. Allez à l'étape 125.	Mode esclave Modbus
125	Lire l'état du port série et vérifier qu'aucune commande n'est active. Réinitialiser le bit de commande TO_CHAR_MODE. <ul style="list-style-type: none"> <li>● S'il n'y a pas d'erreur sur le port série <ul style="list-style-type: none"> <li>○ et si le mode caractère est actif, passer à l'étape 130.</li> <li>○ et le mode caractère n'est pas actif, incrémenter Test_inc et recommencer l'étape 125 jusqu'à 1000 fois. Après 1000 échecs de tentatives, affecter Erreur à 125 et passer à l'étape 130.</li> </ul> </li> <li>● S'il y a une erreur sur le port série, alors <ul style="list-style-type: none"> <li>○ affecter Erreur à 125.</li> <li>○ Allez à l'étape 130.</li> </ul> </li> </ul>	
130	Revenir à l'étape 0.	

## Programmation

### Programmation en langage ST.

L'exemple est programmé en langage ST. La section correspondante se trouve dans la même tâche maître (MAST).

```
CASE Function_Step OF
```

```
0: (* Initialisation *)
```

```
IF (Start) THEN (* indicateur de déclenchement *)
```

```
Error := 0;
```

```
Function_Step := 5; (* étape suivante *)
```

```
END_IF;
```

```
5: (* Envoyer commande pour faire passer le port série du mode Modbus esclave au Mode caractère *)
```

```
READ_STS(Serial_port); (* lire état du port série *)
```

```
IF (Serial_port.EXCH_STS = 0) THEN (* aucune commande active *)
```

```
Serial_port.CONTROL := 16#00; (* réinitialiser mot de commande *)
```

```
(* définir bit de commande TO_CHAR_MODE *)
```

```
SET(Serial_port.TO_CHAR_MODE);
```

```
WRITE_CMD (Serial_port); (* envoyer commande *)
```

```
Test_inc := 0; (* initialiser compteur de nouvelles tentatives *)
```

```
Function_Step := 10; (* étape suivante *)
```

```
END_IF;
```

```
10: (* Tester le résultat de la commande de bascule en Mode caractère*).
```

```
READ_STS(Serial_port); (* lire état du port série *)
```

```
IF (Serial_port.EXCH_STS = 0) THEN (* commande terminée *)
```

```
(* réinitialiser bit de commande TO_CHAR_MODE *)
```

```
RESET(Serial_port.TO_CHAR_MODE);
```

```
IF (Serial_port.EXCH_RPT = 0) THEN (* aucune erreur *)
```

```
IF (AND(Serial_port.PROTOCOL, 16#0F) = 03)
```

```
THEN (* Mode caractère OK *)
```

```
Function_Step := 15; (* étape suivante *)
```

```
ELSE
```

```
Test_inc := Test_inc + 1;
```

```
IF (Test_inc > 1000) THEN
```

```
Error := 10; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
END_IF;
ELSE (* erreur lors de l'envoi de la commande au port *)
Error := 10; (* erreur *)
Function_Step := 130;
END_IF;
END_IF;

15: (* Envoyer commande de numérotation au modem *)
(*Le numéro de téléphone doit être inséré entre 'ATDT' et '$N'*)
ReqString := 'ATDT4001$N'; (* message de numérotation *)
MngtPrint[2] := 500; (* timeout *)
MngtPrint[9] := 9; (* taille de l'échange en octet *)
PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
Function_Step := 20;
20: (* Tester résultat de la fonction PRINT_CHAR *)
IF (NOT MngtPrint[0].0) THEN
IF (MngtPrint[1] = 0) THEN
Function_Step := 25; (* réussi : étape suivante *)
ELSE
Error := 20; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
END_IF;
25: (* Attente de la réponse via INPUT_CHAR *)
MngtInput[2] := 500; (* timeout *)
AnsString1:=' ';
(* attendre réponse du modem *)
INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_connect, MngtInput, AnsString1);
Function_Step := 30; (* étape suivante *)
```

```
30: (* Tester résultat de la fonction INPUT_CHAR *)
IF (NOT MngtInput[0].0) THEN
IF (MngtInput[1] = 0) THEN
Function_Step := 35; (* réussi : étape suivante *)
ELSE
Error := 30; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
END_IF;
```

```
35: (* Tester réponse du modem *)
IF (AnsString1 = '$NCONNET') THEN
Function_Step := 40; (* réussi : étape suivante *)
ELSE
Error := 35; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
```

```
40: (* Envoyer commande pour faire passer le port série du Mode caractère au mode Modbus maître *)
READ_STS(Serial_port); (* lire état du port série *)
IF (Serial_port.EXCH_STS = 0) THEN (* aucune commande active *)
Serial_port.CONTROL := 16#00; (* réinitialiser mot de commande *)
(* définir bit de commande TO_MODBUS_MASTER *)
SET(Serial_port.TO_MODBUS_MASTER);
WRITE_CMD (Serial_port); (* envoyer commande *)
Test_inc := 0; (* initialiser compteur de nouvelles tentatives *)
Function_Step := 45; (* étape suivante *)
END_IF;
```

```
45: (* Tester le résultat de la commande de bascule en mode Modbus maître*).
READ_STS(Serial_port); (* lire état du port série *)
IF (Serial_port.EXCH_STS = 0) THEN (* commande terminée *)
(* bit de commande TO_MODBUS_MASTER *)
```

```
RESET(Serial_port.TO_MODBUS_MASTER);
IF (Serial_port.EXCH_RPT = 0) THEN (* aucune erreur *)
IF (AND(Serial_port.PROTOCOL, 16#0F) = 06)
THEN (* mode Modbus maître OK *)
Function_Step := 50; (* étape suivante *)
ELSE
Test_inc := Test_inc + 1;
IF (Test_inc > 1000) THEN
Error := 45; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
END_IF;
ELSE (* erreur lors de l'envoi de la commande au port *)
Error := 45; (* erreur *)
Function_Step := 130;
END_IF;
END_IF;

50: (*Ecrire des informations dans la deuxième UC*)
Mngtwrite[2]:=50; (* timeouts*)
%MW40:=5; (* valeur à envoyer *)
WRITE_VAR(ADD(Adr_modbus),'%MW',100,2,%MW40:2,Mngtwrite);
Function_Step := 55;

55: (* Tester résultat de la fonction WRITE_VAR *)
IF (NOT Mngtwrite[0].0) THEN
IF (Mngtwrite[1] = 0) THEN
Function_Step := 60; (* réussi : étape suivante *)
ELSE
Error := 55; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
END_IF;
```

```

60: (* Envoyer commande pour basculer le port série du mode Modbus au Mode caractère *)
  READ_STS(Serial_port); (* lire état du port série *)
  IF (Serial_port.EXCH_STS = 0) THEN (* aucune commande active *)
    Serial_port.CONTROL := 16#00; (* réinitialiser mot de commande *)
    (* définir bit de commande TO_CHAR_MODE *)
    SET(Serial_port.TO_CHAR_MODE);
    WRITE_CMD (Serial_port); (* envoyer commande *)
    Test_inc := 0; (* initialiser compteur de nouvelles tentatives *)
    Function_Step := 65; (* étape suivante *)
  END_IF;

```

```

65: (* Tester résultat de la commande de bascule *)
  READ_STS(Serial_port); (* lire état du port série *)
  IF (Serial_port.EXCH_STS = 0) THEN (* commande terminée *)
    (* réinitialiser bit de commande TO_CHAR_MODE *)
    RESET(Serial_port.TO_CHAR_MODE);
    IF (Serial_port.EXCH_RPT = 0) THEN (* aucune erreur *)
      IF (AND(Serial_port.PROTOCOL, 16#0F) = 03)
        THEN (* Mode caractère OK *)
          Function_Step := 70; (* étape suivante *)
        ELSE
          Test_inc := Test_inc + 1;
          IF (Test_inc > 1000) THEN
            Error := 65; (* erreur *)
            Function_Step := 130; (* étape suivante = fin *)
          END_IF;
        ELSE IF;
          ELSE (* erreur lors de l'envoi de la commande au port *)
            Error := 65; (* erreur *)
            Function_Step := 130; (* étape suivante = fin *)
          END_IF;
        END_IF;

```

```
70: (* Raccrochage du modem : étape 1*)
ReqString := '+++'; (* séquence d'échappement *)
MngtPrint[3] := 3; (* taille de l'échange en octet *)
PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
Function_Step := 75; (* étape suivante *)

75: (* Tester résultat de la fonction PRINT_CHAR *)
IF (NOT MngtPrint[0].0) THEN
IF (MngtPrint[1] = 0) THEN
(* Réussi : étape suivante *)
Function_Step := 80;
ELSE
(* Fin sur erreur *)
Error := 75;
Function_Step := 130;
END_IF;
END_IF;

80:
MngtInput[2] := 50; (* timeout *)
INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_ok, MngtInput, AnsString2);
(*Attendre réponse du modem*)
Function_Step := 85; (* étape suivante *)

85: (* Tester résultat de la fonction INPUT_CHAR *)
IF (NOT MngtInput[0].0) THEN
IF (MngtInput[1] = 0) THEN
(* Réussi : étape suivante *)
Function_Step := 90;
ELSE
(* Fin sur erreur *)
Error := 85;
Function_Step := 130;
END_IF;
END_IF;
```

```

90: (* Tester réponse du modem *)
IF (AnsString2 = '$NOK') THEN
Function_Step := 95; (* réussi : étape suivante *)
ELSE
Error := 90; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
95: (* Raccrochage du modem : étape 2*)
ReqString := 'ATH0$N'; (* message de raccrochage *)
MngtPrint[3] := 3; (* taille de l'échange en octet *)
PRINT_CHAR(ADDM(Adr_Char), ReqString, MngtPrint);
Function_Step := 100; (* étape suivante *)
100: (* Tester résultat de la fonction PRINT_CHAR *)
IF (NOT MngtPrint[0].0) THEN
IF (MngtPrint[1] = 0) THEN
(* Réussi : étape suivante *)
Function_Step := 105;
ELSE
(* Fin sur erreur *)
Error := 100;
Function_Step := 130;
END_IF;
END_IF;
105:
MngtInput[2] := 50; (* timeout *)
INPUT_CHAR(ADDM(Adr_Char), 1, nb_charac_to_receive_ok, MngtInput, AnsString3);
(*Attendre réponse du modem*)
Function_Step := 110; (* étape suivante *)
110: (* Tester résultat de la fonction INPUT_CHAR *)
IF (NOT MngtInput[0].0) THEN
IF (MngtInput[1] = 0) THEN
(* Réussi : étape suivante *)
Function_Step := 115;
ELSE

```



```
(* Fin sur erreur *)
Error := 110;
Function_Step := 130;
END_IF;
END_IF;
115: (* Tester réponse du modem *)
IF (AnsString3 = '$NOK') THEN
Function_Step := 120; (* réussi : étape suivante *)
ELSE
Error := 115; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
120: (* Envoyer commande pour basculer le port série du Mode caractère au mode Modbus
esclave *)
READ_STS(Serial_port); (* lire état du port série *)
IF (Serial_port.EXCH_STS = 0) THEN (* aucune commande active *)
Serial_port.CONTROL := 16#00; (* réinitialiser mot de commande *)
(* définir bit de commande TO_MODBUS_SLAVE *)
SET(Serial_port.TO_MODBUS_SLAVE);
WRITE_CMD (Serial_port); (* envoyer commande *)
Test_inc := 0; (* initialiser compteur de nouvelles tentatives *)
Function_Step := 125; (* étape suivante *)
END_IF;

125: (* Tester résultat de la commande de bascule *)
READ_STS(Serial_port); (* lire état du port série *)
IF (Serial_port.EXCH_STS = 0) THEN (* commande terminée *)
(* réinitialiser bit de commande TO_MODBUS_SLAVE *)
RESET(Serial_port.TO_MODBUS_SLAVE);
IF (Serial_port.EXCH_RPT = 0) THEN (* aucune erreur *)
IF (AND(Serial_port.PROTOCOL, 16#0F) = 07)
THEN (* Mode caractère OK *)
Function_Step := 130; (* étape suivante *)
ELSE
```

```
Test_inc := Test_inc + 1;
IF (Test_inc > 1000) THEN
Error := 125; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
END_IF;
ELSE (* erreur lors de l'envoi de la commande au port *)
Error := 125; (* erreur *)
Function_Step := 130; (* étape suivante = fin *)
END_IF;
END_IF;
130: (* Fin *)
IF (NOT Start) THEN (* indicateur de déclenchement *)
Function_Step := 0; (* atteindre état d'attente *)
END_IF;
END_CASE;
```

---

# Chapitre 10

## Démarrage de l'application

---

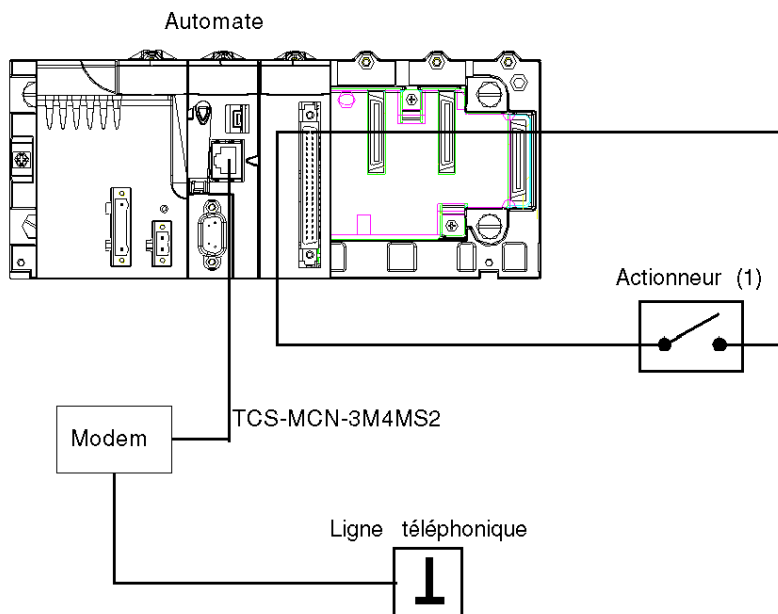
### Exécution de l'application en mode Standard

#### Présentation

Dans cet exemple, l'exécution en mode standard requiert l'utilisation de deux automates, d'un module d'entrée TOR, d'un module BMX NOM 0200 et de 2 modems SR2MOD01.

#### Câblage du premier automate esclave

Le premier automate esclave est raccordé comme suit :



(1) : l'actionneur est connecté à la voie 0 du module TOR.

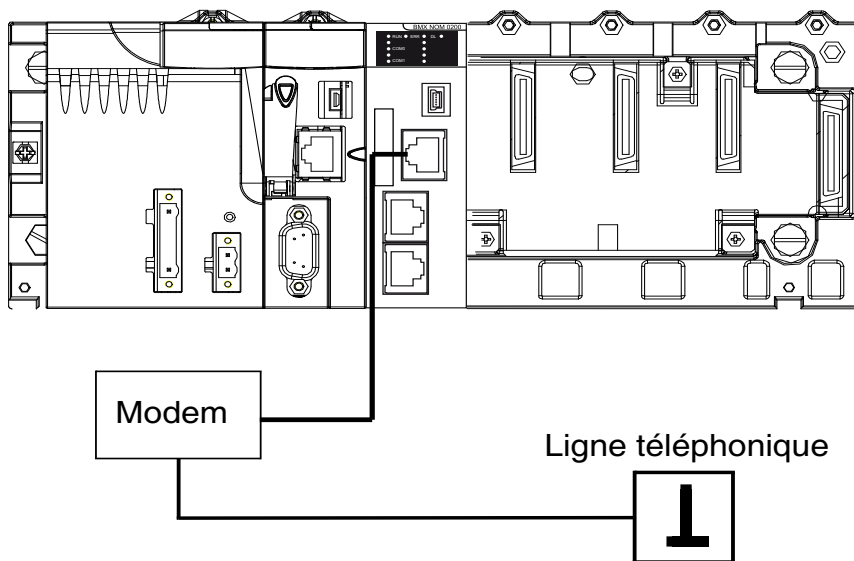
Dans cet exemple, le premier modem est connecté au port série du processeur du premier automate esclave.

L'état de l'actionneur commande l'état de la variable Démarrage dans l'application.

### Câblage du second automate esclave

Le deuxième automate esclave est raccordé comme suit :

#### Automate esclave



Dans cet exemple, le deuxième modem est connecté à la voie 0 du module BMX NOM 0200 du deuxième automate esclave.

Afin de fiabiliser la communication, le câble TCS XCN 3M4F3S4 est utilisé pour gérer les signaux de modem DTR/DSR/DCD par l'application.

### Configuration du deuxième automate esclave

Avant de transférer le projet pour configurer le deuxième automate esclave, vérifiez que le deuxième automate esclave n'est pas connecté au modem.

Le tableau ci-dessous présente la procédure à suivre pour transférer le projet en mode standard :

Etape	Action
1	Dans le menu <i>Automate</i> , cliquez sur <i>Mode Standard</i> ,
2	Dans le menu <i>Génération</i> , cliquez sur <i>Regénérer tout le projet</i> . Votre projet est généré et prêt à être transféré à l'automate.
3	Dans le menu <i>Automate</i> , cliquez sur <i>Connexion</i> . Vous êtes connecté à l'automate.
4	Dans le menu <i>Automate</i> , cliquez sur <i>Transfert du projet vers l'automate</i> . La fenêtre <i>Transfert du projet vers l'automate</i> s'ouvre. Cliquez sur <i>Transférer</i> . L'application est transférée vers l'automate.
5	Raccordez le deuxième automate esclave à un modem SR2MOD01.

### Transfert de l'application au premier automate esclave

Avant de transférer l'application, vérifiez que le premier automate esclave n'est pas raccordé au modem.

Le tableau ci-dessous présente la procédure à suivre pour transférer l'application en mode standard :

Etape	Action
1	Dans le menu <i>Automate</i> , cliquez sur <i>Mode Standard</i> ,
2	Dans le menu <i>Génération</i> , cliquez sur <i>Regénérer tout le projet</i> . Votre projet est généré et prêt à être transféré à l'automate. Lors de la génération du projet, une fenêtre de résultats s'affiche. Si une erreur survient dans le programme, <i>Control Expert</i> indique son emplacement (lorsque vous cliquez sur la séquence mise en évidence).
3	Dans le menu <i>Automate</i> , cliquez sur <i>Connexion</i> . Vous êtes connecté à l'automate.
4	Dans le menu <i>Automate</i> , cliquez sur <i>Transfert du projet vers l'automate</i> . La fenêtre <i>Transfert du projet vers l'automate</i> s'ouvre. Cliquez sur <i>Transférer</i> . L'application est transférée vers l'automate.

### Exécution de l'application sur le premier automate esclave

Le tableau ci-dessous présente la procédure à suivre pour exécuter l'application en mode standard :

Etape	Action
1	Dans le menu <code>Automate</code> , cliquez sur <code>Exécuter</code> . La fenêtre <code>Exécuter</code> s'ouvre. Cliquez sur <code>OK</code> . L'application est maintenant en cours d'exécution sur l'automate.
2	Déconnectez le PC exécutant le logiciel <code>Control Expert</code> du premier automate esclave.
3	Raccordez le premier automate esclave à un modem <code>SR2MOD01</code> .



## !

### %I

Selon la norme CEI, %I indique un objet langage de type entrée TOR.

### %IW

Selon la norme CEI, %IW indique un objet langage de type entrée analogique.

### %KW

Selon la norme CEI, %KW indique un objet langage de type mot constant.

### %M

Selon la norme CEI, %M indique un objet langage de type bit mémoire.

### %MW

Selon la norme CEI, %MW indique un objet langage de type mot mémoire.

### %Q

Selon la norme CEI, %Q indique un objet langage de type sortie TOR.

### %QW

Selon la norme CEI, %QW indique un objet langage de type sortie analogique.

## A

### Adresse

Sur un réseau, l'identification d'une station. Dans une trame, groupement de bits identifiant la source ou la destination de la trame.

### Altivar

Variateur CA à vitesse variable.

### ARRAY

Un ARRAY est une table contenant des éléments de même type. Sa syntaxe est la suivante : ARRAY [<limits>] OF <Type>. Exemple : ARRAY [1..2] OF BOOL est une table à une dimension contenant deux éléments de type BOOL. ARRAY [1..10, 1..20] OF INT est une table à deux dimensions contenant 10 x 20 éléments de type INT.

### ASCII

Abréviation de American Standard Code for Information Interchange (Code standard américain pour l'échange des données). Il s'agit d'un code américain (devenu par la suite un standard international) qui utilise sept bits pour définir chaque caractère alphanumérique utilisé en anglais, les symboles de ponctuation, certains caractères graphiques et d'autres commandes diverses.

## B

### BOOL

BOOL est l'abréviation du type booléen. Il s'agit du type de données de base en informatique. Une variable de type BOOL peut avoir l'une des deux valeurs suivantes : 0 (FALSE) ou 1 (TRUE). Un bit extrait d'un mot est de type BOOL. Par exemple : %MW10.4.

### BYTE

Lorsque huit bits sont regroupés, on parle d'octet (type BYTE). La saisie d'un BYTE s'effectue en mode binaire ou en base 8. Le type BYTE est codé dans un format huit bits qui, au format hexadécimal, s'étend de 16#00 à 16#FF.

## C

### Concentrateur

Appareil concentrateur reliant plusieurs modules centralisés et flexibles afin de créer un réseau.

### Configuration

La configuration recueille les données qui caractérisent la machine (invariant) et nécessaires pour le fonctionnement du module. Toutes ces informations sont stockées dans la zone constante %KW de l'automate. L'application de l'automate ne peut pas les modifier.

### Control Expert

Logiciel de programmation d'automate Schneider Automation.

### CRC

CRC est l'abréviation de Cyclic Redundancy Checksum : elle indique si des caractères ont été « déformés » lors de la transmission des trames.

## D

### DFB

DFB est l'acronyme de « Derived Function Block » (bloc fonction dérivé). Les types DFB sont des blocs de fonctions pouvant être définis par l'utilisateur en langage ST (texte structuré), IL (liste d'instructions), LD (langage à contacts) ou FBD (diagramme de blocs fonction). L'utilisation de ces types DFB dans une application permet :

- de simplifier la conception et la saisie du programme ;
- d'accroître la lisibilité du programme ;
- de faciliter sa mise au point ;
- de diminuer le volume de code généré.

### Diffusion

Les communications de diffusion envoient des paquets d'une station à chacune des destinations du réseau. Les messages de diffusion appartiennent à chaque appareil réseau ou à un seul appareil dont l'adresse n'est pas connue.



## DINT

DINT est l'acronyme du format Double INTeger (entier double) (codé sur 32 bits). Les limites supérieure/inférieure sont les suivantes :  $-2$  à la puissance 31) à  $(2$  à la puissance 31) - 1.  
Exemple :-2147483648, 2147483647, 16#FFFFFFFF.

## E

### EBOOL

EBOOL est l'acronyme du type Extended BOOLEan (booléen étendu). Une variable de type EBOOL possède une valeur (0 pour FALSE ou 1 pour TRUE), mais également des fronts montants ou descendants et des fonctions de forçage. Elle occupe un octet de mémoire. L'octet contient les informations suivantes :

- un bit pour la valeur ;
- un bit pour l'historique (chaque fois que l'objet change d'état, la valeur est copiée dans ce bit) ;
- un bit pour le forçage (égal à 0 si l'objet n'est pas forcé, égal à 1 s'il est forcé).

La valeur par défaut de chaque bit est 0 (FALSE).

### EF

EF est l'acronyme de « Elementary Function » (fonction élémentaire). Il s'agit d'un bloc utilisé dans un programme, qui exécute une fonction logique prédéterminée. Une fonction ne dispose pas d'informations sur l'état interne. Plusieurs appels de la même fonction à l'aide des mêmes paramètres d'entrée fournissent toujours les mêmes valeurs de sortie. Vous trouverez des informations sur la forme graphique de l'appel de fonction dans le « [bloc fonction (instance)] ». Contrairement aux appels de bloc fonction, les appels de fonction ne comportent qu'une sortie qui n'est pas nommée et dont le nom est identique à celui de la fonction. En langage FBD, chaque appel est indiqué par un [numéro] unique via le bloc graphique. Ce numéro est géré automatiquement et ne peut pas être modifié. Vous positionnez et paramétrez ces fonctions dans votre programme afin d'exécuter votre application. Vous pouvez également développer d'autres fonctions à l'aide du kit de développement SDKC.

## F

### FBD

FBD est l'acronyme de « Function Block Diagram » (langage en blocs fonction). FBD est un langage de programmation graphique qui fonctionne comme un logigramme. Par l'ajout de blocs logiques simples (ET, OU, etc.), chaque fonction ou bloc fonction du programme est représenté sous cette forme graphique. Pour chaque bloc, les entrées se situent à gauche et les sorties à droite. Les sorties des blocs peuvent être liées aux entrées d'autres blocs afin de former des expressions complexes.

### Fipio

Bus terrain utilisé pour brancher des appareils de type capteur ou actionneur.

### **Full duplex**

Méthode de transmission de données capable d'envoyer et de recevoir simultanément sur la même voie.

## I

### **INT**

INT est l'acronyme du format « single INTEger » (entier simple) (codé sur 16 bits). Les limites supérieure/inférieure sont les suivantes : - (2 puissance 15) à (2 puissance 15) - 1. Exemple : - 32768, 32767, 2#1111110001001001, 16#9FA4.

### **IODDT**

IODDT est l'acronyme de « Input/Output Derived Data Type » (type de données dérivées d'E/S). Cet acronyme désigne un type de données structuré représentant un module ou une voie d'un module automate. Chaque module expert possède ses propres IODDT.

## L

### **LED**

LED est l'abréviation de Light emitting diode ou diode émettrice de lumière. Voyant qui s'allume lorsque l'électricité le traverse. Signale l'état de fonctionnement d'un module de communication.

### **LRC**

LRC est l'abréviation de Longitudinal redundancy check : ce contrôle de redondance longitudinale a été conçu pour répondre à la faible probabilité de détection des erreurs de contrôle de parité.

## M

### **Mémoire FLASH**

Type de mémoire non volatile (rémanente) susceptible d'être écrasée par écriture. Elle est stockée dans une mémoire EEPROM spéciale, effaçable et reprogrammable.

### **Module TOR**

Module avec entrées/sorties tout ou rien.

### **Momentum**

Modules d'entrées/sorties utilisant plusieurs réseaux de communication standard ouverts.

## P

### **PLC**

PLC est l'abréviation de Programmable logic controller ou automate programmable. Cerveau d'un processus de fabrication industriel. Il automatise le processus, par opposition aux systèmes de contrôle à relais. Les automates programmables sont des ordinateurs conçus pour résister aux conditions parfois difficiles de l'environnement industriel.

## Protocole

Définit les formats de message et un jeu de règles utilisé par au moins deux équipements pour communiquer en utilisant ces formats.

## R

### Réseau

Il existe deux significations du mot "réseau".

- En LD (langage à contacts) : un réseau est un ensemble d'éléments graphiques interconnectés. La portée d'un réseau est locale, par rapport à l'unité (la section) organisationnelle du programme dans laquelle le réseau est situé.
- Avec des modules de communication experts : Un réseau est un groupe de stations qui communiquent entre elles. Le terme « réseau » est également utilisé pour définir un groupe d'éléments graphiques interconnectés. Ce groupe constitue ensuite une partie d'un programme qui peut être composée d'un groupe de réseaux.

### RS232

Norme de communication série qui définit la tension du service suivant :

- un signal de +12 V indique un 0 logique
- un signal de -12 V indique un 1 logique

Cependant, en cas d'atténuation du signal, une détection est fournie jusqu'aux limites -3 V et +3 V. Entre ces deux limites, le signal est considéré comme non valide. Les connexions RS232 sont très sensibles aux interférences. La norme précise de ne pas dépasser une distance de 15 m ou 9 600 bauds (bits/s).

### RS485

Norme de connexion série qui fonctionne dans un différentiel de 10 V/+5 V. Deux fils sont utilisés pour l'envoi et la réception. Leurs sorties "3 états" leur permettent de passer en mode d'écoute une fois la transmission terminée.

### RTU

RTU est l'abréviation de Remote Terminal Unit ou terminal distant. En mode RTU, les données sont envoyées sous forme de deux caractères hexadécimaux de quatre bits, assurant un débit supérieur au mode ASCII pour le même débit en bauds. Modbus RTU est un protocole binaire et plus sensible au décalage temporel que le protocole ASCII.

## S

### Section

Module programmable appartenant à une tâche pouvant être écrit dans le langage choisi par le programmeur (FBD, LD, ST, IL ou SFC). Une tâche peut être composée de plusieurs sections, l'ordre d'exécution des sections au sein de la tâche correspondant à l'ordre dans lequel elles sont créées. Cet ordre peut être modifié.

### **Semi-duplex**

Méthode de transmission de données permettant la communication dans les deux sens, mais dans un seul sens à la fois.

### **SEPAM**

Relais de protection numérique pour la protection, le contrôle et la surveillance des systèmes d'alimentation.

### **Socket**

Association d'un port à une adresse IP, servant d'identification de l'émetteur ou du destinataire.

### **ST**

ST est l'abréviation de Structured Text (littéral structuré). Le langage littéral structuré est un langage élaboré proche des langages de programmation informatiques. Il permet de structurer des suites d'instructions.

### **STRING**

Une variable de type STRING est une chaîne de caractères ASCII. La longueur maximale d'une chaîne est de 65 534 caractères.

## **T**

### **Tâche**

Ensemble de sections et de sous-programmes, exécutés de façon cyclique ou périodique pour la tâche MAST, ou périodique pour la tâche FAST. Une tâche possède un niveau de priorité, et des entrées et des sorties de l'automate lui sont associées. Ces E/S sont actualisées en conséquence.

### **Tâche maître**

Tâche principale du programme. Elle est obligatoire et est utilisée pour effectuer le traitement séquentiel de l'automate.

### **TAP**

TAP est l'abréviation de Transmission Access Point : l'unité de connexion du bus.

### **Trame**

Une trame est un groupe de bits constituant un bloc distinct d'informations. Les trames contiennent des informations ou des données de contrôle de réseau. La taille et la composition d'une trame sont définies par la technique de réseau utilisée.

## **U**

### **UC**

UC vient de l'anglais CPU, l'abréviation de Central Processing Unit : nom générique utilisé pour les processeurs Schneider Electric.

## V

### Variable

Entité mémoire de type BOOL, WORD, DWORD, etc., dont le contenu peut être modifié par le programme en cours d'exécution.

## W

### WORD

Le type WORD est codé dans un format de 16 bits et utilisé pour les traitements sur des chaînes de bits.

Le tableau ci-dessous donne les limites inférieure/supérieure des bases qui peuvent être utilisées :

Base	Limite inférieure	Limite supérieure
Hexadécimale	16#0	16#FFFF
Octale	8#0	8#177777
Binaire	2#0	2#1111111111111111

Exemples de représentation :

Données	Représentation dans l'une des bases
0000000011010011	16#D3
1010101010101010	8#125252
0000000011010011	2#11010011

## X

### XBT

Terminal opérateur graphique.

### XPS

Module de sécurité utilisé pour le traitement des signaux de sécurité qui surveille à la fois le composant et le câblage d'un système de sécurité, avec des périphériques de surveillance générale, ainsi que des modèles spécifiques d'application.





## A

accessoires de câblage, *36*

## B

BMXNOM0200, *17*  
BMXP341000, *17*  
BMXP342000, *17*  
BMXP342010, *17*  
BMXP3420102, *17*  
BMXP342020, *17*  
bus Modbus, *45*

## C

Câblage, *36*  
certifications, *23*  
changement de protocole, *119*  
configuration des paramètres, *87*  
configuration Modbus, *53*  
configuration, mode caractère, *73*

## E

équipements de connexion, *25*

## I

INPUT\_BYTE, *82*  
INPUT\_CHAR, *82*

## M

mise au point Modbus, *67*  
mise au point, mode caractère, *84*  
mise en route, *123*  
mode caractère, *69*

## N

normes, *23*

## P

PRINT\_CHAR, *82*  
programmation, bus Modbus, *62*  
programmation, mode caractère, *82*

## S

structure des données de voie pour communication Modbus  
    T\_COM\_MB\_BMX, *102*  
structure des données de voie pour communication Mode caractère  
    T\_COM\_CHAR\_BMX, *110, 111*  
structure des données de voie pour communications Modbus  
    T\_COM\_MB\_BMX, *103*  
structure des données de voie pour les protocoles de communication  
    T\_COM\_STS\_GEN, *97*  
Structure des données de voie pour les protocoles de communication  
    T\_COM\_STS\_GEN, *98*  
structure des données de voie pour tous les modules  
    T\_GEN\_MOD, *116*  
structures des données de voie pour tous les modules  
    T\_GEN\_MOD, *116*

## T

T\_COM\_CHAR\_BMX, *110, 111*  
T\_COM\_MB\_BMX, *102, 103*  
T\_COM\_STS\_GEN, *97, 98*  
T\_GEN\_MOD, *116, 116*

