

Modicon M340 avec Unity Pro

Liaison série
Manuel utilisateur

12/2015

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.

Aucune partie de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique ou photocopie, sans autorisation préalable de Schneider Electric.

Toutes les réglementations de sécurité pertinentes locales doivent être observées 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.

© 2015 Schneider Electric. Tous droits réservés.

Table des matières



	Consignes de sécurité	9
	A propos de ce manuel	11
Partie I	Présentation des communications Modbus Serial et Mode caractère	13
Chapitre 1	Présentation des communications Modbus Serial et Mode caractère	15
	Présentation des communications Modbus Serial et Mode caractère.	15
Partie II	Mise en œuvre matérielle des communications Modbus Serial et Mode caractère	17
Chapitre 2	Introduction aux communications série	19
2.1	Liaison série sur les processeurs BMX P34 1000/2000/2010/2020 . .	20
	Présentation de la liaison série sur les processeurs BMX P34 1000/2000/2010/20102/2020.	20
2.2	Module BMX NOM 0200 à deux ports RS-485/232.	25
	Présentation du module BMX NOM 0200 à 2 ports RS-485/232	26
	Équipement Modicon M340H (renforcé)	32
	Mise à la terre des modules installés	33
	Installation du module BMX NOM 0200	35
	Points à prendre en compte pour le câblage du BMX NOM 0200. . . .	37
Chapitre 3	Architectures de communications série	39
3.1	Architectures de communications série des processeurs BMX P34 1000/2000/2010/20102/2020	40
	Terminaison de ligne Modbus et polarisation (RS485)	41
	Raccordement des équipements Modbus (RS485).	43
	Raccordement d'un équipement terminal de données (DTE) (RS232)	46
	Raccordement d'un équipement terminal de circuit de données (DCE) (RS232)	48
3.2	Architectures de communications série du BMX NOM 0200.	51
	Terminaison de ligne Modbus et polarisation (RS485)	52
	Raccordement des équipements Modbus (RS485).	54
	Raccordement d'un équipement terminal de données (DTE) (RS232)	56
	Raccordement d'un équipement terminal de circuit de données (DCE) (RS232)	58
3.3	Câblage	61
	Câblage	61

Partie III	Mise en œuvre logicielle des communications	
	Modbus Serial et Mode caractère.....	65
Chapitre 4	Méthodologie de mise en œuvre	67
	Introduction à la phase d'installation	67
Chapitre 5	Communication Modbus Serial des processeurs	
	BMX P34 1000/2000/2010/20102/2020.....	71
5.1	Généralités.....	72
	A propos de Modbus Serial	73
	Performances.....	74
	Comment accéder aux paramètres de la liaison série.....	76
5.2	Configuration d'une communication Modbus Serial.....	79
	Ecran de configuration d'une communication Modbus Serial	80
	Fonctions Modbus accessibles	82
	Valeurs par défaut des paramètres d'une communication Modbus Serial	83
	Paramètres Modbus liés à l'application	84
	Paramètres Modbus liés à la transmission	86
	Paramètres liés aux signaux et à la ligne physique dans Modbus.	88
5.3	Programmation d'une communication Modbus Serial	91
	Services pris en charge par le processeur maître d'une liaison Modbus	92
	Services pris en charge par le processeur esclave d'une liaison Modbus	100
5.4	Mise au point d'une communication Modbus Serial.....	102
	Ecran de mise au point d'une communication Modbus Serial	102
Chapitre 6	Communication en mode caractère des processeurs	
	BMX P34 1000/2000/2010/20102/2020.....	105
6.1	Généralités.....	106
	A propos de la communication en mode caractère	107
	Performances.....	108
6.2	Configuration d'une communication en mode caractère	110
	Ecran de configuration d'une communication en mode caractère	111
	Fonctions accessibles en mode caractère.....	114
	Valeurs par défaut des paramètres d'une communication en mode caractère	115
	Paramètre de détection de fin de message en mode caractère	116
	Paramètres liés à la transmission en mode caractère	118
	Paramètres liés aux signaux et à la ligne physique en mode caractère	120
6.3	Programmation d'une communication en mode caractère.....	122
	Fonctions de communication en mode caractère	122

6.4	Mise au point d'une communication en mode caractère	128
	Ecran de mise au point d'une communication en mode caractère . . .	128
Chapitre 7	Communications Modbus Serial du BMX NOM 0200 .	131
7.1	Généralités	132
	A propos de Modbus Serial	133
	Performances	134
	Comment accéder aux paramètres de la liaison série	136
7.2	Configuration d'une communication Modbus Serial	139
	Ecran de configuration des communications série Modbus dans un rack local Modicon M340	140
	Ecran de configuration de la communication série Modbus du module BMX NOM 0200 dans une station X80	143
	Fonctions Modbus accessibles	146
	Valeurs par défaut des paramètres d'une communication Modbus Serial	147
	Paramètres Modbus liés à l'application	148
	Paramètres Modbus liés à la transmission	150
	Paramètres liés aux signaux et à la ligne physique dans Modbus . . .	152
	Configuration de l'adresse de l'esclave MODBUS BMX NOM0200 sans Unity Pro	154
7.3	Programmation d'une communication Modbus Serial	156
	Services pris en charge par le module maître d'une liaison Modbus .	157
	Services pris en charge par le module esclave d'une liaison Modbus	165
	Informations sur le mode expert Modbus	167
7.4	Mise au point d'une communication Modbus Serial	175
	Ecran de mise au point d'une communication Modbus série	175
Chapitre 8	Communication en Mode caractère du BMX NOM 0200	179
8.1	Généralités	180
	A propos de la communication en mode caractère	180
8.2	Configuration d'une communication en mode caractère	181
	Ecran de configuration de la communication en mode caractère du module BMX NOM 0200 dans un rack local	182
	Ecran de configuration de la communication en mode caractère du module BMX NOM 0200 dans une station X80	185
	Fonctions accessibles en mode caractère	188

	Valeurs par défaut des paramètres de communication en mode caractère	189
	Paramètre de détection de fin de message en mode caractère	190
	Paramètres liés à la transmission en mode caractère	192
	Paramètres liés aux signaux et à la ligne physique en mode caractère	194
8.3	Programmation d'une communication en mode caractère.	197
	Fonctions de communication en mode caractère	198
	Informations sur le mode expert.	207
8.4	Mise au point d'une communication en mode caractère	212
	Ecran de mise au point d'une communication en mode caractère.	212
Chapitre 9	Diagnostics de module BMX NOM 0200	215
9.1	Diagnostics de module BMX NOM 0200	216
	Diagnostics du module BMX NOM 0200	217
	Diagnostic détaillé par voie de communication	219
Chapitre 10	Objets langage des communications Modbus et Mode caractère	221
10.1	Objets langage et IODDT des communications Modbus et Mode caractère	222
	Présentation des objets langage pour les communications Modbus et Mode caractère	223
	Objets langage à échange implicite associés à la fonction métier	224
	Objets langage à échanges explicites associés à la fonction métier	225
	Gestion des échanges et des comptes rendus avec des objets explicites	227
10.2	Objets langage et IODDT génériques pour les protocoles de communication.	230
	Informations détaillées sur les objets à échange implicite de l'IODDT de type T_COM_STS_GEN.	231
	Détails sur les objets à échange explicite de l'IODDT de type T_COM_STS_GEN	232
10.3	Objets langage et IODDT associés aux communications Modbus	234
	Informations détaillées sur les objets langage à échange explicite pour une fonction Modbus	235
	Informations détaillées sur les objets à échanges implicites IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT	236
	Informations détaillées sur les objets à échanges explicites IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT	237
	Informations détaillées sur les objets langage associés au mode de communication Modbus	240

10.4	Objets langage et IODDT associés aux communications Modbus en mode caractère	243
	Informations détaillées sur les objets langage à échange explicite pour la communication en Mode caractère.	244
	Informations détaillées sur les objets à échange implicite IODDT de type T_COM_CHAR_BMX	245
	Informations détaillées sur les objets à échange explicite IODDT de type T_COM_CHAR_BMX	246
	Informations détaillées sur les objets langage associés à la configuration en mode caractère	249
10.5	IODDT de type T_GEN_MOD applicable à tous les modules	253
	Détails des objets langage du IODDT de type T_GEN_MOD	253
10.6	Objets langage et DDT d'équipement associés aux communications Modbus	255
	Noms de DDT d'équipements de communication	255
Chapitre 11	Changement dynamique de protocoles	259
	Changement de protocole avec les processeurs BMX P34 1000/2000/2010/20102/2020	260
	Changement de protocole avec le module BMX NOM 0200.	262
Partie IV	Démarrage rapide : BMX NOM 0200.4	265
Chapitre 12	Présentation	267
	Présentation du produit	268
	Présentation de l'architecture	269
	Limitations	271
Chapitre 13	Configuration dans Unity Pro	273
	Insertion de modules	274
	Ecran de configuration du module.	275
Partie V	Mise en route : Exemple de mise en œuvre d'une liaison série.	279
Chapitre 14	Description de l'application.	281
	Vue d'ensemble de l'application	281
Chapitre 15	Installation de l'application avec Unity Pro	283
15.1	Présentation de la solution utilisée	284
	Différentes étapes du processus utilisant Unity Pro	284

15.2	Développement de l'application	285
	Création du projet	286
	Déclaration des variables	291
	Utilisation d'un modem	295
	Procédure de programmation	297
	Structure de programmation	299
	Programmation	302
Chapitre 16	Démarrage de l'application	311
	Exécution de l'application en mode Standard	311
Glossaire		315
Index		323

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.

A propos de ce manuel



Présentation

Objectif du document

Ce manuel décrit la mise en œuvre matérielle et logicielle des communications Modbus et en mode caractère pour les processeurs BMX P34 1000/2000/2010/20102/2020. Il décrit également l'installation matérielle et logicielle des modules de communication BMX NOM 0200 pour les automates Modicon M340 et les stations X80.

Champ d'application

Cette documentation est applicable à partir de Unity Pro 10.0 ou version ultérieure.

Information spécifique au produit

AVERTISSEMENT

FONCTIONNEMENT IMPREVU 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

Présentation des communications Modbus Serial et Mode caractère

Chapitre 1

Présentation des communications Modbus Serial et Mode caractère

Présentation des communications Modbus Serial et Mode caractère

Général

Les liaisons série pour les processeurs BMX P34 1000/2000/2010/20102/2020 et le module BMX NOM 0200 prennent en charge deux protocoles de communication :

- Modbus Serial
- Mode caractère

Protocole Modbus

Modbus est un protocole standard qui possède les propriétés suivantes :

- définit une communication client/serveur entre différents modules présents au sein d'un bus ou d'une liaison série. Le client est identifié par le maître et les modules esclaves représentent les serveurs.
- s'appuie sur un mode d'échange de données composé de requêtes et de réponses offrant des services à travers différents codes fonction,
- définit un moyen d'échanger des trames d'application de type Modbus selon 2 types de codage :
 - Mode RTU
 - Mode ASCII

La gestion des échanges se fait de la manière suivante :

- 1 seul appareil peut émettre sur le bus,
- les échanges sont gérés par le maître. Seul le maître peut générer des échanges. Aucun esclave ne peut de lui-même envoyer de message sans y avoir été invité.
- le maître réitère la requête lors d'un échange erroné et l'esclave auquel est adressée la requête est déclaré absent par le maître s'il ne répond pas dans un délai donné.
- l'esclave émet une réponse d'exception au maître s'il ne comprend pas le message ou si il ne peut pas le traiter. Dans ce cas, le maître peut réitérer ou non la requête.

2 types de dialogue sont possibles entre maître et esclave(s) :

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

Communication en Mode caractère

Le mode caractère est un mode d'échanges de données point à point entre 2 entités.

Contrairement au protocole Modbus, il ne définit ni des communications série hiérarchisées ni des services à l'aide de codes fonction.

Le mode caractère est un mode asynchrone. Chaque information de texte est émise ou reçue caractère par caractère de façon irrégulière dans le temps. Les propriétés suivantes permettent de délimiter les échanges dans le temps :

- 1 ou 2 caractères de fin de trame,
- timeout,
- nombre de caractères.

Partie II

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
2	Introduction aux communications série	19
3	Architectures de communications série	39

Chapitre 2

Introduction aux communications série

Objet de ce chapitre

Ce chapitre présente les communications sur les processeurs BMX P34 1000/2000/2010/20102/2020 et sur le module BMX NOM 0200.

Le tableau ci-dessous donne une vue générale de deux possibilités de mise en œuvre des communications par liaison série :

Utilisation du port intégré du processeur	Utilisation du module de communication BMX NOM 0200
- Vitesse de transmission limitée - Lignes série non isolées - Fourniture d'alimentation à l'équipement terminal	- Nombre supérieur de voies de communication disponibles - Gestion des signaux RS232 spécifiques du modem - Vitesse de transmission supérieure - Deux lignes série RS485 isolées

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
2.1	Liaison série sur les processeurs BMX P34 1000/2000/2010/2020	20
2.2	Module BMX NOM 0200 à deux ports RS-485/232	25

Sous-chapitre 2.1

Liaison série sur les processeurs BMX P34 1000/2000/2010/2020

Présentation de la liaison série sur les processeurs BMX P34 1000/2000/2010/20102/2020.

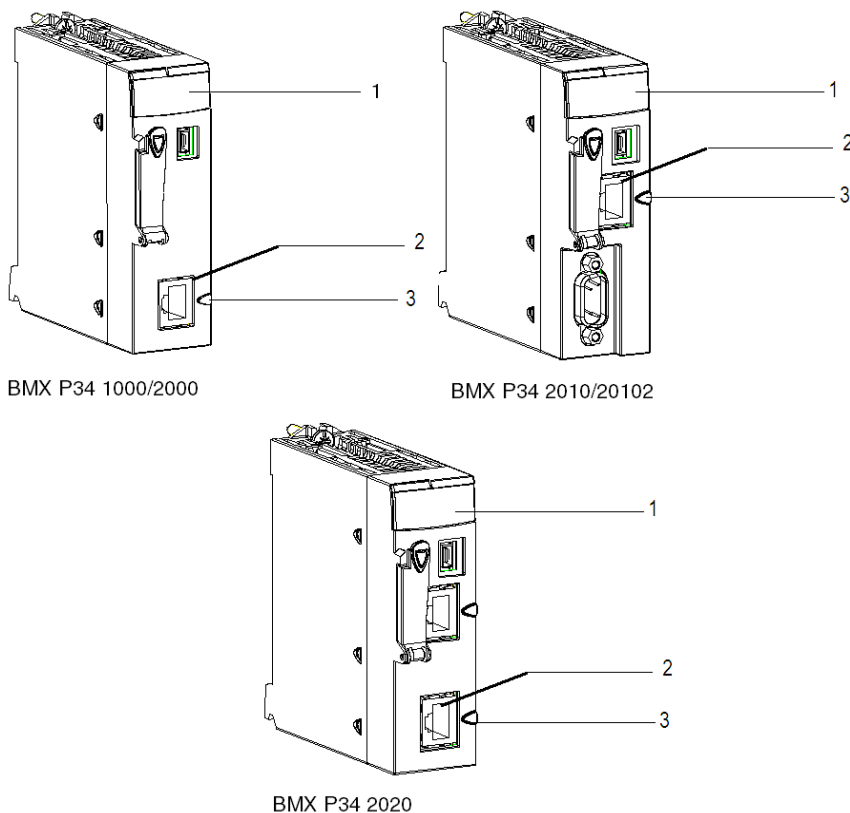
Général

Les processeurs suivants comportent une voie de communication intégrée dédiée aux communications série et permettent la communication par liaison série :

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

Présentation des processeurs

La figure ci-dessous présente les caractéristiques physiques des processeurs BMX P34 1000/2000/2010/20102/2020 :



Ces processeurs sont composés des éléments suivants :

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

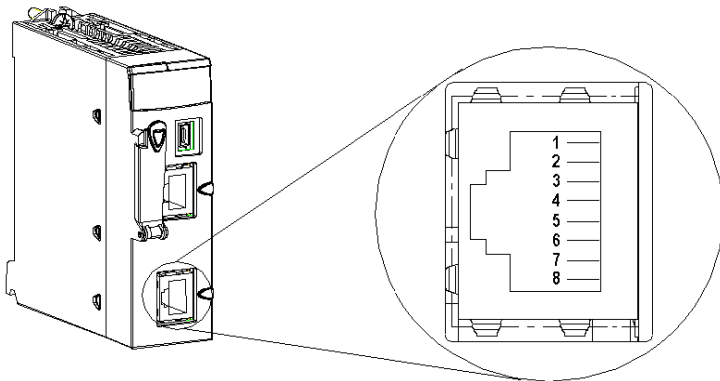
Diagnostic visuel de la communication série

L'état des communications série est indiqué par le voyant orange SER COM à l'avant de ces processeurs :

- Voyant clignotant : la communication série est active.
- Voyant éteint : la communication série est inactive.

Présentation du port série

L'illustration ci-dessous représente le port série RJ45 :



Le connecteur RJ45 comporte huit broches. Les broches utilisées diffèrent selon la liaison physique adoptée.

Les broches utilisées par la liaison série RS232 sont :

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

Les broches utilisées par la liaison série RS485 sont :

- broche 4 : signal D1
- broche 5 : signal D0

La broche 7 est dédiée à l'alimentation des interfaces homme-machine ou des petits équipements par le câble de liaison série :

- broche 7 : alimentation de la liaison série : 5 VCC/90 mA

Détails des caractéristiques

Caractéristiques CC :

- Consommation stabilisée maximum : 190 mA,
- Tension minimale sur connecteur du processeur pour 190 mA : 4,9 V,
- Tension maximale sur connecteur du processeur pour 190 mA : 5,25 V,
- Tension maximale sur connecteur du processeur sans aucune charge : 5,5 V.

Caractéristiques CA :

- Charge du condensateur : (sur 5 V)
 - Condensateur en céramique maximum 1 μ F
 - 10 μ F en tantale
- Démarrage de charge de la pompe : (sur 5 V)
 - 4 condensateurs en céramique 1 μ F
 - 2 condensateurs en tantale 10 μ F

NOTE : les câbles à quatre fils RS232, à deux fils RS485 et à deux fils RS485 avec alimentation utilisent tous le même connecteur femelle RJ45. Seul le câblage des signaux diffère.

Caractéristiques électriques de ligne

Les lignes RS232 et RS485 ne sont pas isolées.

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

La polarisation de la ligne RS485 est intégrée à l'automate. Elle est activée ou désactivée automatiquement par le système, selon la configuration choisie dans l'écran Unity Pro :

- 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 polarisation n'est pas affectée par le changement de protocole dynamique. Les valeurs des résistances de polarisation est de 560 ohms.

En mode RS232, la polarisation n'est pas requise.

Il n'y a pas de terminaison de ligne intégrée.

Caractéristiques de voie

La voie de ces processeurs inclut :

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

Les spécifications de liaison pour les deux protocoles sont :

	Modbus série / RS485	Modbus série / RS232	Mode caractère / RS485	Mode caractère / RS232
Type	Maître/esclave	Maître/esclave	Half Duplex	Full Duplex
Flux	19 200 bauds. Les paramètres peuvent être réglés de 300 bauds à 38 400 bauds.	19 200 bauds. Les paramètres peuvent être réglés de 300 bauds à 38 400 bauds.	9 600 bauds. Les paramètres peuvent être réglés de 300 bauds à 38 400 bauds.	9 600 bauds. Les paramètres peuvent être réglés de 300 bauds à 38 400 bauds.
Nombre d'équipements	32	32	—	—
Adresses d'esclave autorisées	1 à 247	1 à 247	—	—
Longueur maximale du bus sans dérivation	1 000 m (15 m avec dérivation)	15 m	1 000 m (15 m avec dérivation)	15 m
Taille de message	Modbus série : ● RTU : 256 octets (252 octets de données) ● ASCII : 513 octets (2 x 252 octets de données)	Modbus série : ● RTU : 256 octets (252 octets de données) ● ASCII : 513 octets (2 x 252 octets de données)	1 024 octets	1 024 octets
Utilitaires	Lecture de mots/bits Ecriture de mots/bits Diagnostics.	Lecture de mots/bits Ecriture de mots/bits Diagnostics.	Chaînes d'envoi de caractères. Chaînes de réception de caractères.	Chaînes d'envoi de caractères. Chaînes de réception de caractères.

Sous-chapitre 2.2

Module BMX NOM 0200 à deux ports RS-485/232

Objet de cette section

Cette section présente les communications série sur le module BMX NOM 0200.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Présentation du module BMX NOM 0200 à 2 ports RS-485/232	26
Equipement Modicon M340H (renforcé)	32
Mise à la terre des modules installés	33
Installation du module BMX NOM 0200	35
Points à prendre en compte pour le câblage du BMX NOM 0200	37

Présentation du module BMX NOM 0200 à 2 ports RS-485/232

Général

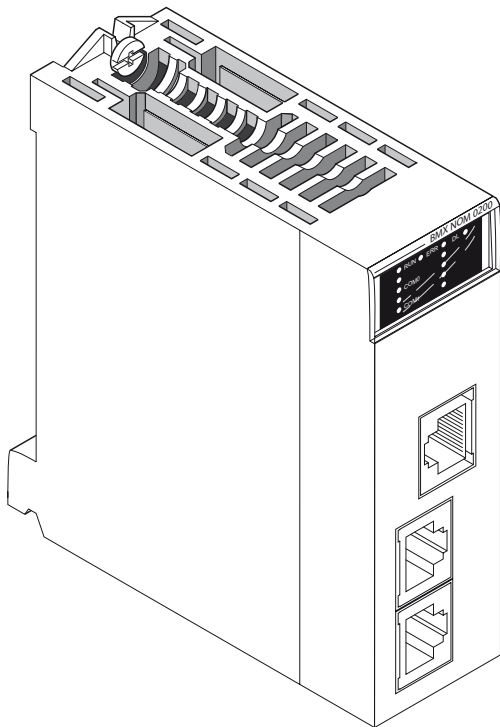
Les modules de liaison série BMX NOM 0200 et BMX NOM 0200H (*voir page 32*) sont des modules de ligne série asynchrones bidirectionnels prenant en charge les communications série Modbus (maître ou esclave) et mode caractère.

Le BMX NOM 0200 est un module dédié de format simple à installer sur le rack de la station Modicon M340.

NOTE : en cas de températures extrêmes (-25 à 0°C et 60 à 70°C, soit -13 à 32°F et 140 à 158°F), les caractéristiques de fonctionnement du BMX NOM 0200H sont les mêmes que celles du BMX NOM 0200 dans la plage de températures normale (0 à 60°C, 32 à 140°F).

Introduction au module

L'illustration ci-dessous présente les caractéristiques physiques du module BMX NOM 0200 :



Ce module BMX NOM 0200 est composé des éléments décrits dans le tableau suivant :

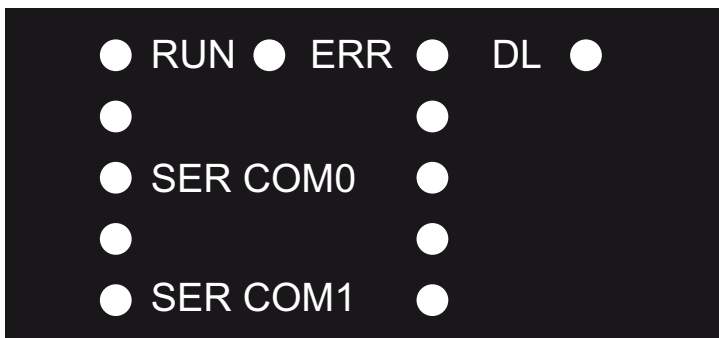
Touche	Description
1	Cinq voyants à l'avant du module : ● RUN et ERR indiquent l'état du module, ● SER COM0 affiche l'état du trafic sur le port 0 ou 1 (voie 0), ● SER COM1 affiche l'état du trafic sur le port 2 (voie 1), ● DL indique l'état du téléchargement du micrologiciel.
2	Voie intégrée (voie 0) dédiée à la liaison série avec 2 ports série : RS232 (port 0) et RS485 (port 1). Remarque : un seul port peut être actif à la fois.
3	Voie intégrée (voie 1) dédiée à la liaison série avec 1 port série : RS485 (port 2).

NOTE : dans certains modes de fonctionnement, des **voyants peuvent donner des informations plus précises** (*voir page 27*).

Diagnostics visuels

Cinq voyants se trouvent sur la face avant du module BMX NOM 0200. Ils affichent des informations sur l'état de fonctionnement du module et sur l'état de communication de la liaison série intégrée.

Affichage des voyants :



- RUN = le module est alimenté et bien configuré.
- ERR = le module a détecté une erreur et ne peut pas fonctionner correctement.
- DL = le micrologiciel est en cours de téléchargement.
- SER COM0 = Communication détectée sur le port 0 ou 1 (voie 0).
- SER COM1 = Communication détectée sur le port 2 (voie 1).

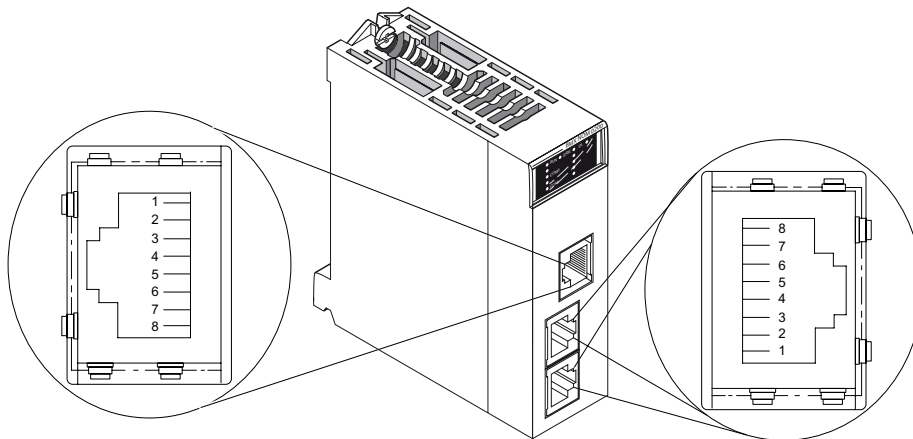
Signification des voyants :

- Chaque témoin LED peut être dans un des états suivants :
 - 1 = Allumé.
 - 0 = Éteint.
 - B = Clignotant.
- Pendant le démarrage du module, tous les voyants sont allumés, puis éteints, pour vérifier qu'ils fonctionnent correctement.

RUN	ERR	SER COM0	SER COM1	DL	Diagnostic
0	–	–	–	–	Le module n'est pas alimenté ou n'est pas opérationnel.
0	B	–	–	–	Le module n'est pas configuré.
1	1	–	–	–	Le module fonctionne mal.
1	1	1	0	–	Le module a détecté une erreur sur la voie 0.
1	1	1	B	–	Le module a détecté un problème sur la voie 0, la voie 1 est en cours d'échange de données.
1	1	0	1	–	Le module a détecté une erreur sur la voie 1.
1	1	B	1	–	Le module a détecté un problème sur la voie 1, la voie 0 est en cours d'échange de données.
1	0	B	–	–	La voie 0 est en cours d'échange de données.
1	0	–	B	–	La voie 1 est en cours d'échange de données.
B	B	–	–	0	L'UC est absente.
B	B	B	B	–	Le module est en cours d'autotest.
–	–	–	–	B	Un micrologiciel de module est en cours de téléchargement.
–	–	–	–	1	Le micrologiciel a été transféré ; le module doit être réinitialisé.

Présentation des ports série

La figure suivante représente les ports série RJ45 du BMX NOM 0200 :



Le tableau ci-dessous indique l'affectation des broches pour le port série sur le BMX NOM 0200 :

N° de broche	RS485 voie 1 / port 1 ou 2	RS232 voie 0 / port 0
1	—	RXD (réception de données)
2	—	TXD (Transmission de données)
3	—	RTS (demande d'émission)
4	D1 (B/B4)	DTR (Terminal de données prêt)
5	D0 (A/A4)	DSR (Equipement de données prêt)
6	—	CTS (Prêt à émettre)
7	—	DCD (Signal détection de porteuse)
8	Mise à la terre de la liaison série potentielle (0 V)	Mise à la terre de la liaison série potentielle (0 V)

NOTE :

- les deux lignes RS485 sont isolées. La tension d'isolement entre les deux lignes série à 500 V et entre chaque ligne isolée de l'embase atteint 500 VCA.
- Les lignes RS232 à sept fils et RS485 à deux fils utilisent le même connecteur RJ45 femelle. Seul le câblage des signaux diffère.

Caractéristiques des voies

Les voies du module BMX NOM 0200 sont notamment :

- Deux interfaces physiques RS485 isolées,
- Une interface physique RS232 non isolée,
- Types de communication Modbus série (ASCII et RTU) et en mode caractère.

Les spécifications de liaison pour les deux protocoles sont :

	Modbus série / RS485	Modbus série / RS232	Mode caractère / RS485	Mode caractère / RS232
Type	Maître/esclave	Maître/esclave	Half Duplex	Full Duplex
Flux	19 200 bauds. Les paramètres peuvent être réglés de 300 bauds à 57 600 bauds.	19 200 bauds. Les paramètres peuvent être réglés de 300 bauds à 115 200 bauds.	9 600 bauds. Les paramètres peuvent être réglés de 300 bauds à 57 600 bauds.	9 600 bauds. Les paramètres peuvent être réglés de 300 bauds à 115 200 bauds.
Nombre d'équipements	32	32	—	—
Adresses d'esclave autorisées	1 à 247	1 à 247	—	—
Longueur maximale du bus sans dérivation	Reportez-vous au tableau ci-dessous (15 m avec dérivation).	15 m	Reportez-vous au tableau ci-dessous (15 m avec dérivation).	15 m
Taille de message	Modbus série : • RTU : 256 octets (252 octets de données) • ASCII : 513 octets (2 x 252 octets de données)	Modbus série : • RTU : 256 octets (252 octets de données) • ASCII : 513 octets (2 x 252 octets de données)	1 024 octets	1 024 octets
Utilitaires	Lecture de mots/bits Ecriture de mots/bits Diagnostics.	Lecture de mots/bits Ecriture de mots/bits Diagnostics.	Chaînes d'envoi de caractères. Chaînes de réception de caractères.	Chaînes d'envoi de caractères. Chaînes de réception de caractères.
Contrôle de flux matériel	—	Facultativement par signaux RTS/CTS.	—	Facultativement par signaux RTS/CTS.

Le tableau présente la longueur maximale de câble RS485 susceptible d'être utilisée, en fonction du débit en bauds choisi :

Débit en bauds (bits/s)	Longueur (m)	Réf. commerciale
300	1000	(1)
600	1000	(1)
1200	1000	(1)
2400	1000	(1)
9600	1000	(1)
19200	600	(1)
38400	300	(1) ou (2)
57600	200	(1) ou (2)

- (1) : Calibre de câbles blindés à paire torsadée AWG24 (TSX CSA 100, TSX CSA 200, TSX CSA 500)
- (2) : Catégorie de câble 5 ou supérieure

Consommation du module BMX NOM 0200

Ce tableau indique la consommation du module BMX NOM 0200 :

Tension	Courant habituel	Courant maximum	Puissance dissipée habituelle	Puissance dissipée maximale
24 VCC	80 mA	130 mA	1,92 W	3,12 W

Équipement Modicon M340H (renforcé)

M340H

Les équipements Modicon M340H sont des versions renforcées des équipements M340. Ils peuvent être utilisés à des températures extrêmes (-25 à 70°C, -13 à 158°F) et dans des environnements chimiques difficiles.

Ce traitement augmente la capacité d'isolation des cartes de circuit et leur résistance :

- à la condensation,
- aux atmosphères poussiéreuses (contenant des particules étrangères conductrices),
- à la corrosion chimique, notamment en cas d'utilisation dans des atmosphères sulfureuses (pétrole, raffinerie, usine de purification, etc.) ou contenant des éléments halogènes (chlorure, etc.).

Dans des conditions de température normales (0 à 60 °C, 32 à 140 °F), les équipements M340H ont les mêmes caractéristiques que les équipements M340 standard.

A des températures extrêmes (-25 à 0°C et 60 à 70°C) (-13 à 32°F et 140 à 158°F), les versions renforcées peuvent avoir des puissances nominales réduites qui affectent les calculs d'alimentation des applications Unity Pro.

Si ces équipements sont utilisés à des températures hors de la plage 25 à 70°C (-13 à 158°F), ils peuvent fonctionner de manière anormale.

ATTENTION

FONCTIONNEMENT D'EQUIPEMENT NON INTENTIONNEL

N'utilisez pas les équipements M340H hors des plages de températures autorisées.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

Les cartes électroniques du matériel renforcé sont dotées d'un revêtement enrobant. Cette protection, associée à une installation et une maintenance appropriées, offre une plus grande robustesse dans des environnements chimiques difficiles.

Mise à la terre des modules installés

Général

La mise à la terre des modules Modicon M340 est indispensable pour éviter tout choc électrique.

Mise à la terre des processeurs et des alimentations

DANGER

RISQUE D'ELECTROCUTION, D'EXPLOSION OU D'ARC ELECTRIQUE

Vérifiez que les contacts de raccordement à la terre sont présents et ne sont pas tordus. S'ils sont absents ou tordus, n'utilisez pas le module et contactez votre représentant Schneider Electric.

Le non-respect de ces instructions provoquera la mort ou des blessures graves.

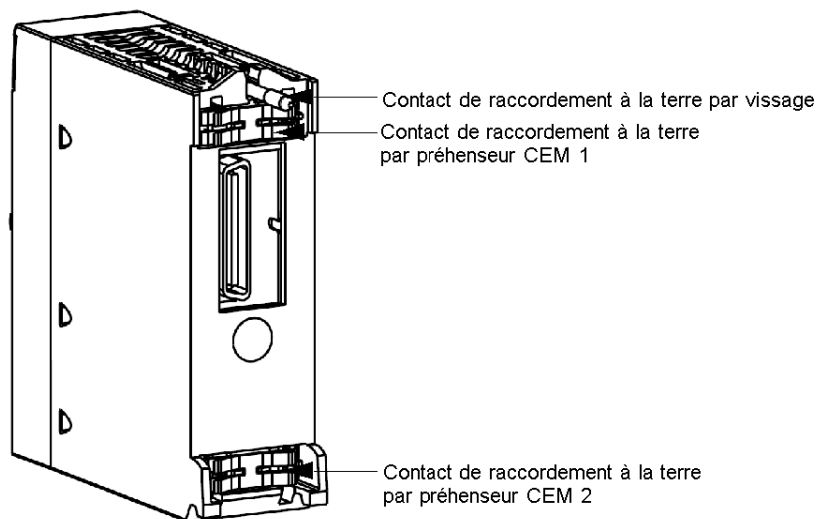
AVERTISSEMENT

COMPORTEMENT IMPREVU DE L'EQUIPEMENT

Serrez les vis des modules. Une rupture dans le circuit peut entraîner un comportement inattendu du système.

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

Tous les modules Modicon M340 possèdent des contacts de terre en face arrière pour la mise à la terre.



Ces contacts raccordent le bus de mise à la terre des modules au bus de mise à la terre du rack.

Installation du module BMX NOM 0200

Général

Le module BMX NOM 0200 est installé dans un rack de station Modicon M340 et ne peut pas utiliser les emplacements réservés à l'alimentation et au processeur. Cette installation doit être conforme aux instructions d'installation du rack.

Le module BMX NOM 0200 nécessite l'installation d'un module CPU dont le système d'exploitation est de version 02.10 au minimum. Cette installation doit être conforme aux instructions fournies.

Un connecteur RJ45 peut ensuite être connecté au module selon le réseau visé.

NOTE : le module BMX NOM 0200 peut être installé dans un rack alors que l'application s'exécute sur l'automate.

AVERTISSEMENT

FONCTIONNEMENT D'EQUIPEMENT NON INTENTIONNEL

L'utilisation de ce produit requiert la maîtrise de la conception et de la programmation des systèmes de contrôle. Seules les personnes ayant 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.

Nombre de modules

Comme le nombre de voies expertes gérées par une station d'automate dépend du processeur installé, le nombre maximum de modules BMX NOM 0200 dans une station est donc influencé par :

- le nombre de voies configurées sur chaque module BMX NOM 0200 (chaque voie compte comme une voie experte),
- le type et la version du processeur installé (*voir Modicon M340 sous Unity Pro, Processeurs, racks et modules d'alimentation, Manuel de configuration*),
- le nombre de voies expertes déjà utilisées.

Lors de la génération de l'application, Unity Pro vérifie que la limitation n'est pas dépassée.

Connexion/Déconnexion

Le module BMX NOM 0200 peut être branché ou débranché sous tension. Quand le module est débranché du rack, sa mémoire interne est effacée. Il passe par une phase d'initialisation lors de sa reconnexion à l'embase.

Un NOM0200 (à partir de la version 1.2) peut être inséré dans un rack sur un emplacement libre sans configuration préalable. Cela est très pratique pour relier une PUNIT quand l'UC n'est pas configurée ou pour ajouter un point de connexion. Dans ce cas, le BMX NOM0200 présente une configuration par défaut.

La configuration par défaut du BMX NOM 0200 est un esclave MODBUS à l'adresse 248, une RTU (délai entre les trames = 2 ms), 8 bits de données, 1 bit d'arrêt, une parité paire, RS232 à 115 200 bits/s sur la voie 0 et RS485 à une vitesse de 57 600 bits/s sur la voie 1.

L'adresse 248 est l'adresse point à point à laquelle répond n'importe quel module esclave BMX NOM 0200. Cette fonctionnalité vise à connecter directement tout module esclave dont l'adresse est inconnue.

Mise à jour du micrologiciel

Le micrologiciel du BMX NOM 0200 peut être mis à jour par le fond de panier de l'automate. La mise à jour du micrologiciel est définie dans le *Manuel utilisateur de Unity Loader, un logiciel SoCollaborative*.

Points à prendre en compte pour le câblage du BMX NOM 0200

Considérations opérationnelles

AVERTISSEMENT

COMPORTEMENT IMPREVU DE L'EQUIPEMENT

Même si vous pouvez connecter ou déconnecter les câbles du module BMX NOE 0200 et des UC BMX P34 20x0 alors que la station BMX XBP est sous tension, cette opération risque d'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 en cours d'alimentation.
- Les modules sont réinitialisés lorsqu'ils sont remis sous tension.

Chapitre 3

Architectures de communications série

Objet de ce chapitre

Ce chapitre présente les architectures utilisant la communication série des processeurs BMX P34 1000/2000/2010/20102/2020 et du module BMX NOM 0200 ainsi que les exigences de câblage.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
3.1	Architectures de communications série des processeurs BMX P34 1000/2000/2010/20102/2020	40
3.2	Architectures de communications série du BMX NOM 0200	51
3.3	Câblage	61

Sous-chapitre 3.1

Architectures de communications série des processeurs BMX P34 1000/2000/2010/20102/2020

Objet de cette section

Cette section présente les architectures utilisant la communication série des processeurs BMX P34 1000/2000/2010/20102/2020 ainsi que le câblage à mettre en œuvre.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Terminaison de ligne Modbus et polarisation (RS485)	41
Raccordement des équipements Modbus (RS485)	43
Raccordement d'un équipement terminal de données (DTE) (RS232)	46
Raccordement d'un équipement terminal de circuit de données (DCE) (RS232)	48

Terminaison de ligne Modbus et polarisation (RS485)

Présentation

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

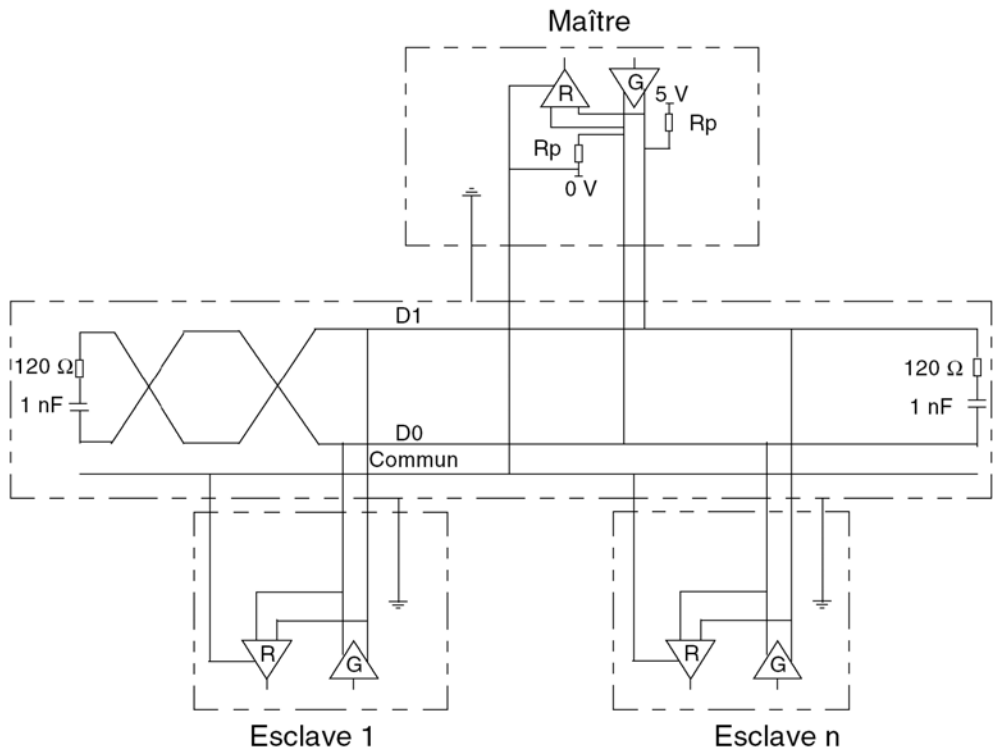
Les équipements susceptibles d'être connectés à ce bus sont :

- 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 44*) 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 Ω et d'un condensateur de 1 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 23)** 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 Ω (R_p) connectées sur la paire équilibrée RS485 :
 - une résistance de démarrage à une tension de 5 V sur le circuit D1,
 - une résistance d'arrêt sur le circuit commun D0.

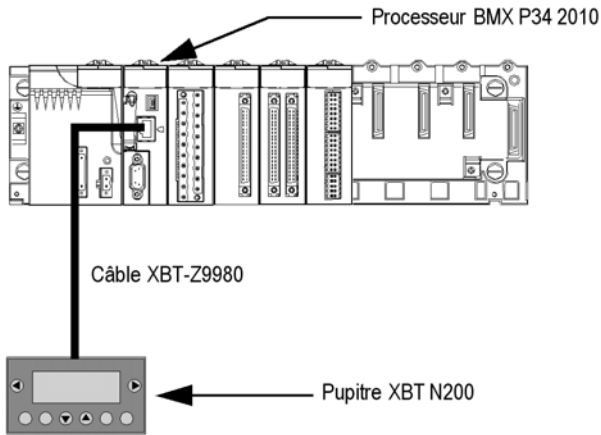
Raccordement des équipements Modbus (RS485)

Général

Les pages ci-dessous présentent deux exemples de branchement d'équipements Modbus et une architecture de liaison Modbus Serial.

Branchement d'équipements Modbus alimentés par la liaison série

L'illustration ci-dessous montre comment connecter un processeur BMX P34 2010 à une console XBT N200 alimentée par la liaison série Modbus :



Les équipements sont configurés de la manière suivante :

- processeur BMX P34 2010 configuré en esclave,
- interface homme machine XBT N200 configurée en maître.

Le câble XBT-Z9980 a les caractéristiques suivantes :

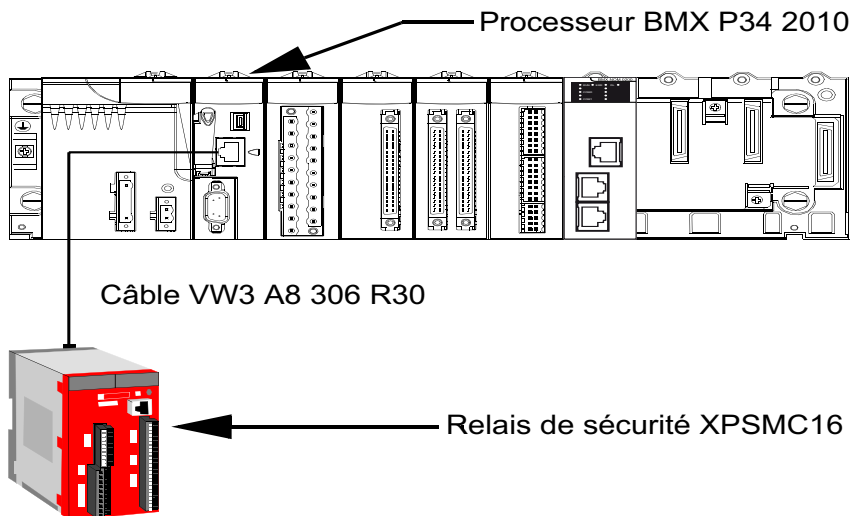
- Connexion : 2 connecteurs mâles RJ45
- Câblage : 2 fils pour la ligne physique RS485 et 2 fils pour l'alimentation liaison série.

Branchement d'équipements Modbus non alimentés par la liaison série

Cette architecture est composée des éléments suivants :

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

La figure suivante représente le raccordement d'un processeur BMX P34 2010 à un relais de sécurité XPSMC16 :



Les équipements sont configurés de la manière suivante :

- processeur BMX P34 2010 configuré en maître,
- relais de sécurité XPSMC16 configuré en esclave.

Le câble VW3 A8 306 R30 a les caractéristiques suivantes :

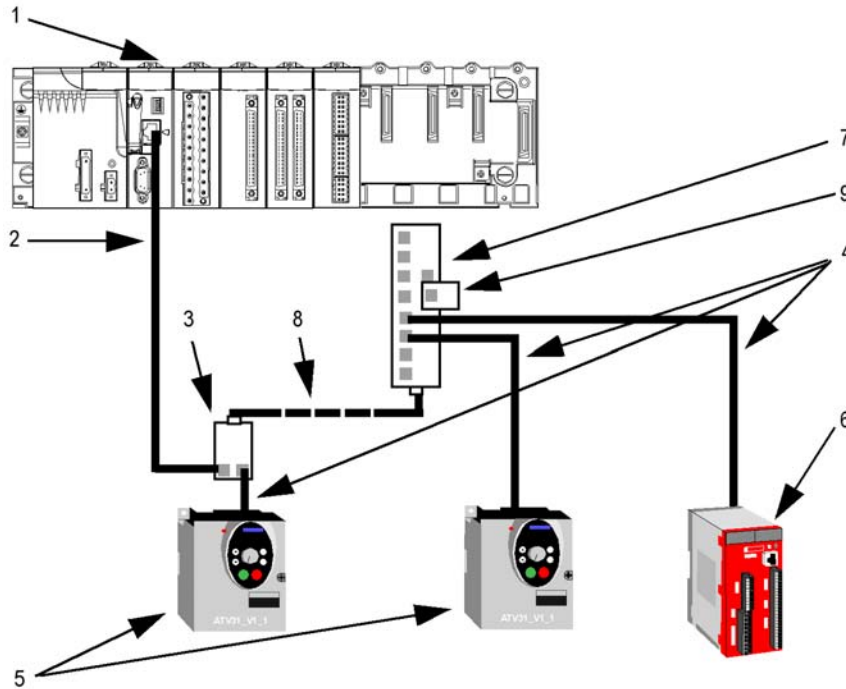
- Connexion : 2 connecteurs mâles RJ45
- Câblage : 2 fils pour la ligne physique RS485.

Architecture de liaison série Modbus

L'architecture de liaison série Modbus est composée des éléments suivants :

- un processeur BMX P34 2010/20102 configuré en maître,
- un relais de sécurité XPSMC16 configuré en esclave,
- un répartiteur isolé TWDXCAISO,
- un répartiteur LU9GC3,
- 2 variateurs ATV31 configurés en esclave.

La figure suivante représente l'architecture de liaison série décrite ci-dessus :



- 1 Processeur BMX P34 2010
- 2 Câble XBT-Z9980
- 3 Répartiteur isolé TWDXCAISO
- 4 Câble VW3 A8 306 R30
- 5 Variateur ATV31
- 6 Relais de sécurité XPSMC16
- 7 Répartiteur LU9 GC3
- 8 Câble TSXCSAx00
- 9 Câble VW3 A8 306 R

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

Général

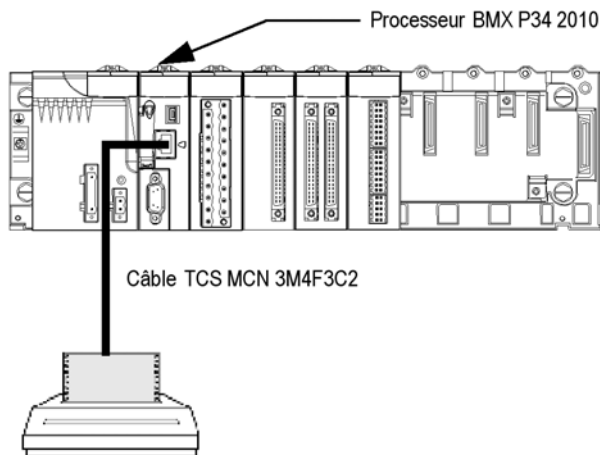
Les équipements terminaux de données sont des équipements tels que :

- 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.

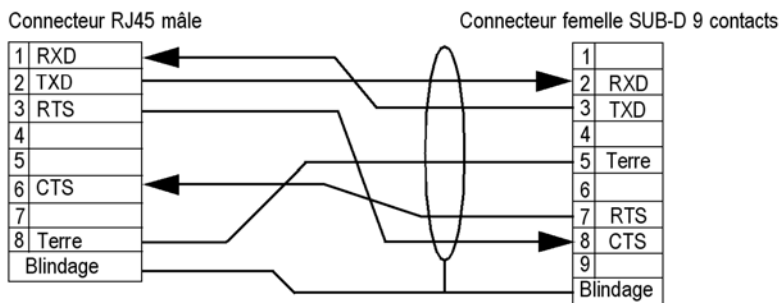
NOTE : Un seul équipement terminal de données peut être raccordé à un processeur BMX P34 1000/2000/2010/20102/2020.

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	● Câble TCS MCN 3M4F3C2 ● Adaptateur TSX CTC 07
Connecteur femelle SUB-D 25 contacts	● Câble TCS MCN 3M4F3C2 ● Adaptateur TSX CTC 10

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.

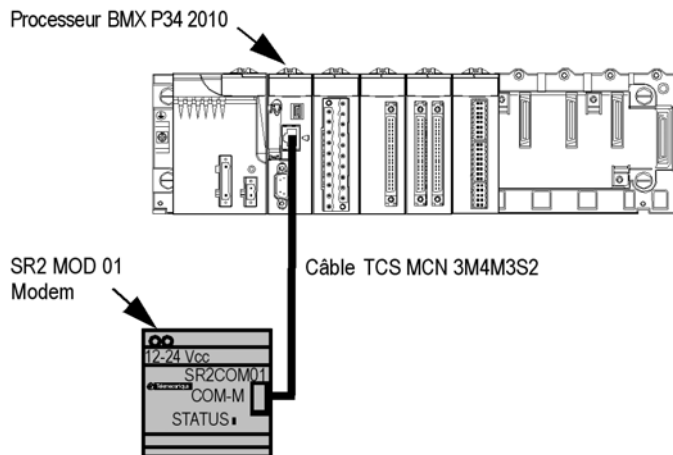
Caractéristiques d'un modem

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 :



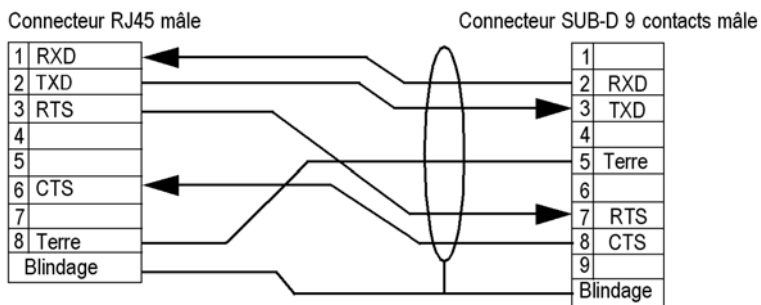
NOTE : En Modbus Serial, le délai d'attente doit être compris entre 100 et 250 ms.

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 :

Connecteur série de l'équipement connecté de données	Câblage
Connecteur femelle SUB-D 9 broches	Câble TCS MCN 3M4M3S2
Connecteur femelle SUB-D 25 broches	● Câble TCS MCN 3M4M3S2 ● Adaptateur TSX CTC 09

Sous-chapitre 3.2

Architectures de communications série du BMX NOM 0200

Objet de cette section

Cette section présente les architectures utilisant la communication série du module BMX NOM 0200 ainsi que le câblage à mettre en œuvre.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Terminaison de ligne Modbus et polarisation (RS485)	52
Raccordement des équipements Modbus (RS485)	54
Raccordement d'un équipement terminal de données (DTE) (RS232)	56
Raccordement d'un équipement terminal de circuit de données (DCE) (RS232)	58

Terminaison de ligne Modbus et polarisation (RS485)

Présentation

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

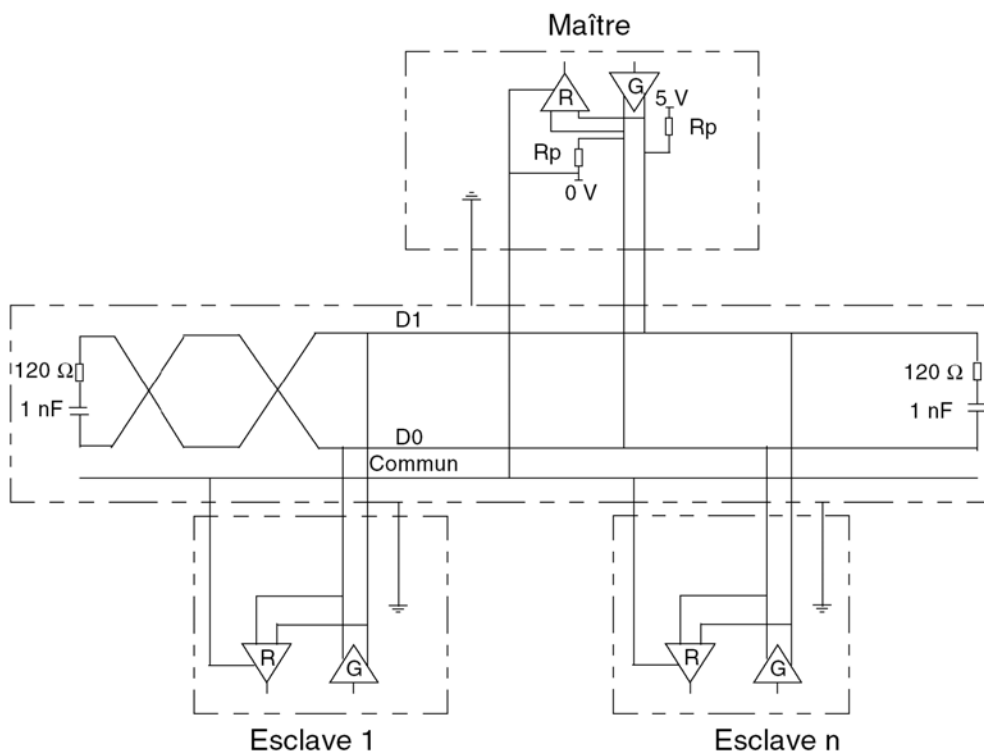
Les équipements susceptibles d'être connectés à ce bus sont :

- 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 54) incluant un processeur BMX NOM 0200 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 Ω et d'un condensateur 1 nF placés sur chacune des extrémités du réseau (VW3 A8 306RC 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 le module BMX NOM 0200 est utilisée comme maître, il est automatiquement piloté par le système donc il n'y a pas besoin de polarisation externe.
- Si le module BMX NOM 0200 est utilisé comme un esclave, la polarisation doit être mise en œuvre par deux résistances de 450 à 650 Ω (Rp) connectées sur la paire équilibrée RS485 :
 - une résistance de démarrage à une tension de 5 V sur le circuit D1,
 - une résistance d'arrêt sur le circuit commun D0.

NOTE :

en mode caractère, la polarisation de la ligne est configurable sous Unity Pro. Il est possible de choisir entre :

- polarisation à basse impédance comme sur les réseaux Modbus (l'objectif de ce type de polarisation est de permettre au maître de maintenir l'état par défaut),
- polarisation à haute impédance (l'objectif de ce type de polarisation est de permettre à chaque appareil de contribuer au maintien de l'état par défaut),
- pas de polarisation (en cas d'utilisation de polarisation externe).

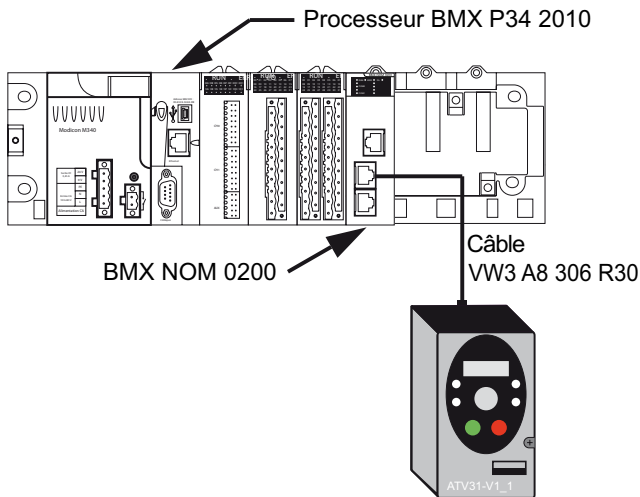
Raccordement des équipements Modbus (RS485)

Général

Les pages ci-dessous présentent un exemple de branchement d'équipement Modbus et une architecture de liaison Modbus Serial.

Branchement d'équipements Modbus non alimentés par la liaison série

L'illustration suivante présente le raccordement d'un module BMX NOM 0200 à un variateur ATV31 :



Les équipements sont configurés de la manière suivante :

- un processeur BMX P34 2010,
- un module BMX NOM 0200 configuré en maître,
- un variateur ATV31 configuré en esclave.

Le câble VW3 A8 306 R30 a les caractéristiques suivantes :

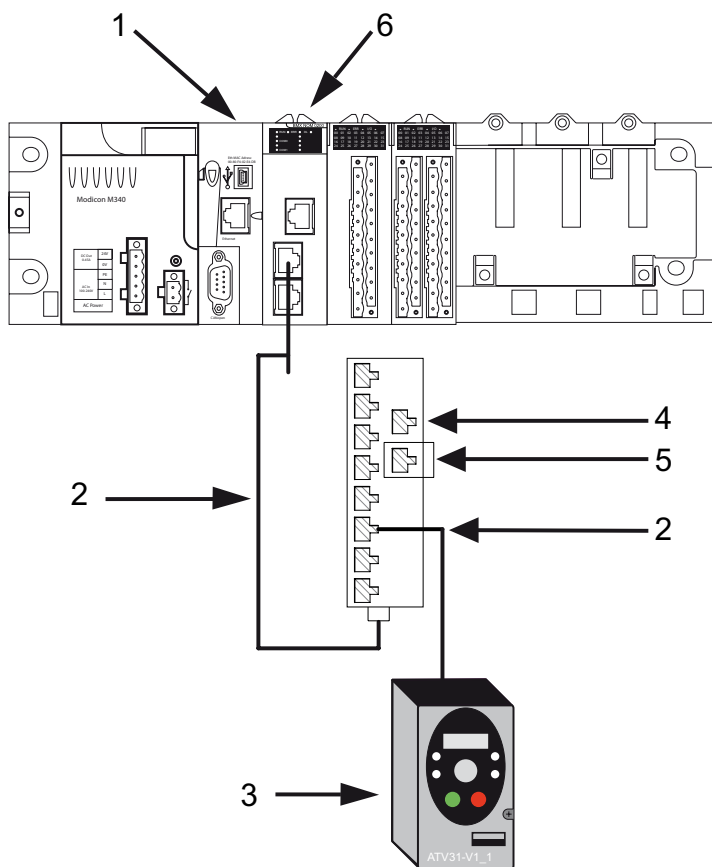
- Connexion : 2 connecteurs mâles RJ45
- Câblage : 2 fils pour la ligne physique RS485.

Architecture de liaison série Modbus

L'architecture de liaison série Modbus est composée des éléments suivants :

- un processeur BMX P34 2010,
- un module BMX NOM 0200 configuré en maître,
- un répartiteur isolé TWDXCAISO,
- un répartiteur LU9GC3,
- deux variateurs ATV31 configurés en esclave.

La figure suivante représente l'architecture de liaison série décrite ci-dessus :



- 1 Processeur BMX P34 2010
- 2 Câble VW3 A8 306 R30
- 3 Variateur ATV31
- 4 Répartiteur LU9 GC3
- 5 Câble VW3 A8 306 R
- 6 Module BMX NOM 0200

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

Général

Les équipements terminaux de données sont des équipements tels que :

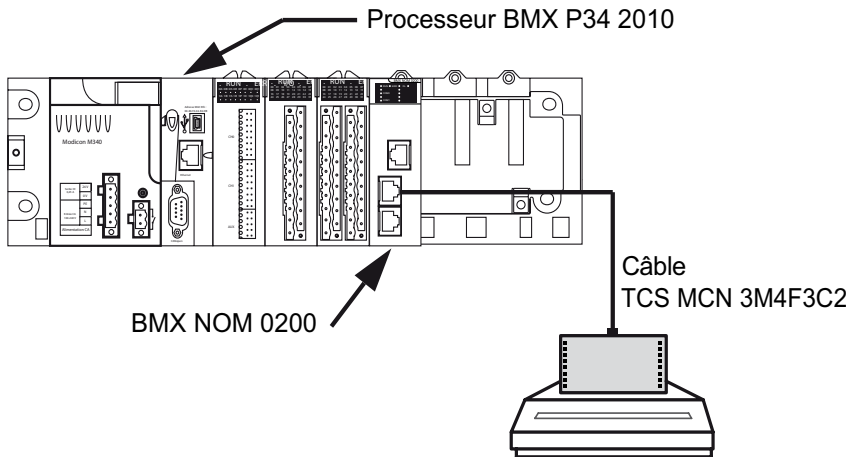
- périphériques usuels (imprimante, écran-clavier, terminal d'atelier...),
- périphériques spécialisés (lecteur de codes barres par exemple),
- ordinateur PC.

Pour un appareil de type DTE, les broches RTS et CTS sont croisées.

Tous les équipements terminaux de données sont raccordés au module BMX NOM 0200 par un câble série croisé utilisant la ligne physique RS232.

Raccordement d'un équipement terminal de données

La figure suivante présente le raccordement d'une imprimante à un module BMX NOM 0200 :



La communication utilisée est une communication en Mode caractère.

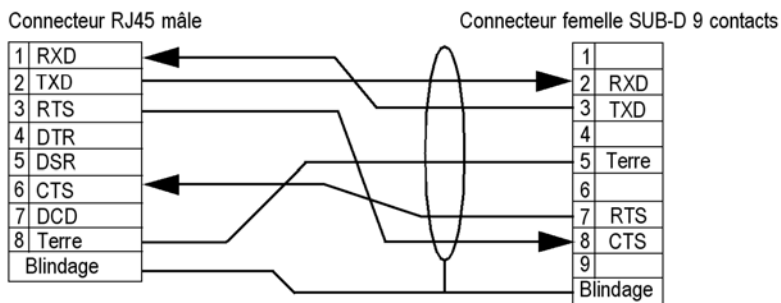
NOTE : Un seul équipement terminal de données peut être raccordé à un module BMX NOM 0200.

Câble série croisé RS 232

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

- RJ45 mâle,
- SUB-D 9 broches 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 SUB-D 9 contacts mâle	Câble TCS MCN 3M4F3C2
Connecteur SUB-D 25 contacts mâle	• Câble TCS MCN 3M4F3C2 • Adaptateur TSX CTC 07
Connecteur femelle SUB-D 25 broches	• Câble TCS MCN 3M4F3C2 • Adaptateur TSX CTC 10

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 circuit de données sont raccordés au module BMX NOM 0200 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

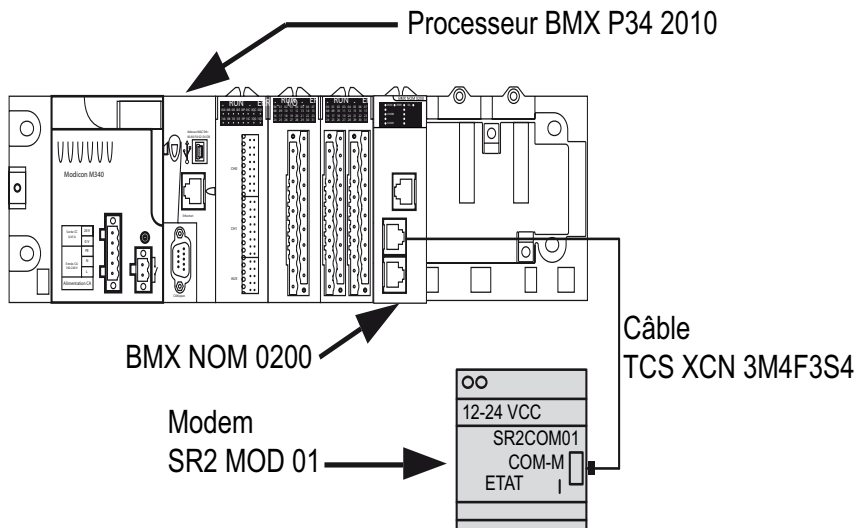
Le module BMX NOM 0200 fonctionne avec la plupart des modems du marché. Pour raccorder un modem au port série d'un module BMX NOM 0200, ce modem doit impérativement avoir les caractéristiques suivantes :

- 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.

Les signaux CTS, DTR, DSR et DCD peuvent être gérés par l'application.

Raccordement d'un équipement terminal de circuit de données

La figure suivante présente le raccordement d'un modem à un module BMX NOM 0200 :



La connexion au modem nécessite un câble modem spécifique pour fonctionner.

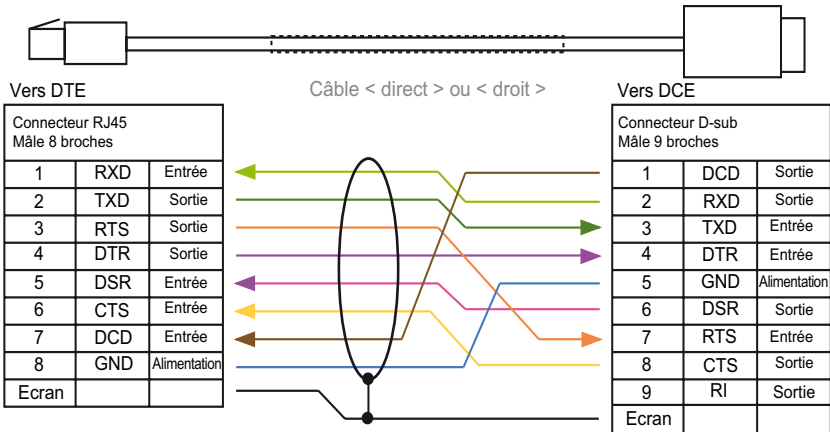
Câble série direct RS 232

Exemple d'un câble TCS XCN 3M4F3S4 :

Le câble droit série TCS XCN 3M4F3S4 est une version à 8 fils et il a deux connecteurs :

- RJ45 mâle,
- SUB-D 9 broches mâle.

La figure suivante représente le brochage d'un câble série direct TCS XCN 3M4F3S4 :



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 :

Connecteur série de l'équipement connecté de données	Câblage
Connecteur femelle SUB-D 9 broches	• Câble TCS MCN 3M4M3S2 • Câble TCS XCN 3M4F3S4
Connecteur femelle SUB-D 25 broches	• Câble TCS MCN 3M4M3S2 • Adaptateur TSX CTC 09

Sous-chapitre 3.3

Câblage

Câblage

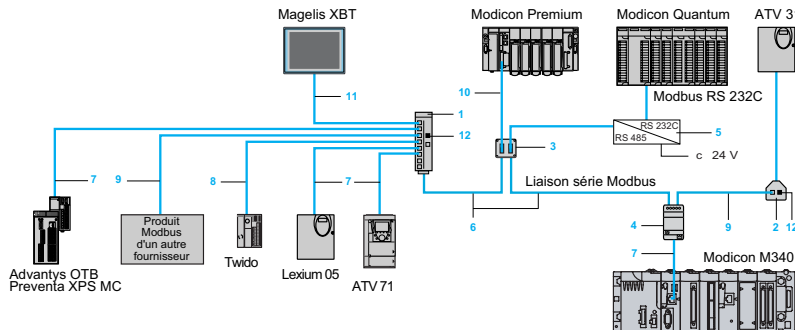
Général

Plusieurs câbles et accessoires sont nécessaires pour établir une liaison série sur les processeurs et le module suivants :

- BMX P34 1000,
- BMX P34 2000,
- BMX P34 2010/20102,
- BMX P34 2020,
- module BMX NOM 0200.

Système de câblage

La figure ci-dessous présente un exemple de système de câblage de liaison Modbus Serial et mode caractère Modicon M340. Les **câbles** (*voir page 62*) et **accessoires de connexion** (*voir page 63*) référencés sur la figure sont décrits dans les tableaux suivants :



Câbles

Le tableau suivant indique les câbles disponibles compatibles avec les communications série sur ces processeurs et ce module :

Référence de figure	Désignation	Longueur	Caractéristiques	Réf. commerciale
6	Câble principal double RS485 paire torsadée blindée	100 m	2 extrémités dénudées	TSX CSA 100
6	Câble principal double RS485 paire torsadée blindée	200 m	2 extrémités dénudées	TSX CSA 200
6	Câble principal double RS485 paire torsadée blindée	500 m	2 extrémités dénudées	TSX CSA 500
7	Câble Modbus RS485	0,3 m	2 connecteurs mâles RJ45,	VW3 A8 306 R03
7	Câble Modbus RS485	1 m	2 connecteurs mâles RJ45,	VW3 A8 306 R10
7	Câble Modbus RS485	3 m	2 connecteurs mâles RJ45,	VW3 A8 306 R30
-	Câble Modbus RS485	3 m	● 1 connecteur mâle RJ45 ● 1 connecteur mâle SUB-D 15 broches	VW3 A8 306
4	Câble Modbus RS485	0,3 m	● 1 connecteur mâle RJ45 ● Un connecteur mini-DIN	TWD XCA RJ003
4	Câble Modbus RS485	1 m	● 1 connecteur mâle RJ45 ● Un connecteur mini-DIN	TWD XCA RJ010
4	Câble Modbus RS485	3 m	● 1 connecteur mâle RJ45 ● Un connecteur mini-DIN	TWD XCA RJ030
5	Câble Modbus RS485	3 m	● 1 connecteur mâle RJ45 ● 1 extrémité dénudée	VW3 A8 306 D30
9	Câble Modbus RS485	3 m	● Un connecteur miniature ● Un connecteur SUB-D 15 broches	TSX SCP CM 4630
11	Câble RS485 pour écran et terminal Magelis XBT	2,5 m	● 1 connecteur mâle RJ45 ● 1 connecteur femelle SUB-D 25 broches Remarque : Ce câble n'est pas compatible avec le module BMX NOM 0200	XBT-Z938
-	Câble RS485 pour appareils alimentés par la liaison série	3 m	2 connecteurs mâles RJ45, Remarque : Ce câble n'est pas compatible avec le module BMX NOM 0200.	XBT-Z9980

Référence de figure	Désignation	Longueur	Caractéristiques	Réf. commerciale
-	Câble RS 232 4 fils pour équipements terminaux de données (DTE)	3 m	1 connecteur mâle RJ45 1 connecteur femelle SUB-D 9 broches	TCS MCN 3M4F3C2
-	Câble RS 232 4 fils pour équipements terminaux de circuit de données (DCE)	3 m	1 connecteur mâle RJ45 1 connecteur mâle SUB-D 9 broches	TCS MCN 3M4M3S2
-	Câble RS232 7 fils pour équipements terminaux de circuit de données (DCE)	3 m	1 connecteur mâle RJ45 1 connecteur mâle SUB-D 9 broches	TCS XCN 3M4F3S4

Accessoires de raccordement

Le tableau suivant indique les accessoires de connexion disponibles compatibles avec les communications série sur ces processeurs et ce module :

Référence de figure	Désignation	Caractéristiques	Réf. commerciale
1	Boîte de répartition Modbus	10 connecteurs RJ45 1 bornier à vis	LU9 GC3
2	Boîte de dérivation en T	2 connecteurs RJ45 - Câble intégré de 0,3 m avec connecteur RJ45 à son extrémité	VW3 A8 306 TF03
2	Boîte de dérivation en T	2 connecteurs RJ45 - Câble intégré de 1 m avec connecteur RJ45 à son extrémité	VW3 A8 306 TF10
-	Boîte de dérivation en T passive	3 borniers à vis - adaptateur fin de ligne RC	TSX SCA 50
3	Prise d'abonné 2 voies passive	2 connecteurs mâles SUB-D 25 broches 2 borniers à vis - adaptateur fin de ligne RC	TSX SCA 62
4	Boîte de dérivation en T RS485 isolée	- Un connecteur RJ45 1 bornier à vis	TWD XCA ISO
-	Boîte de dérivation en T	3 connecteurs RJ45	TWD XCA T3RJ

Référence de figure	Désignation	Caractéristiques	Réf. commerciale
-	Adaptateur Modbus / Bluetooth	● Un adaptateur Bluetooth avec un connecteur RJ45 ● Un jeu de cordons pour PowerSuite avec deux connecteurs RJ45 ● Un jeu de cordons pour TwidoSuite avec un connecteur RJ45 et un connecteur mini-DIN ● Un adaptateur RJ45/SUB-D mâle 9 broches pour les variateurs de vitesse ATV	TWD XCA T3RJ
5	Adaptateur de ligne RS232C/RS485 sans signaux de modem	19,2 kbit/s	XGS Z24
12	Terminaison de ligne pour connecteur RJ45	● Résistance de 120 Ω ● Capacité de 1 nF	VW3 A8 306 RC
-	Terminaison de ligne pour bornier à vis	● Résistance de 120 Ω ● Capacité de 1 nF	VW3 A8 306 DRC
-	Adaptateur pour équipements non standard	● 2 connecteurs mâles SUB-D 25 broches	XBT ZG999
-	Adaptateur pour équipements non standard	● 1 connecteur mâle SUB-D 25 broches ● 1 connecteur mâle SUB-D 9 broches	XBT ZG909
-	Adaptateur pour équipements terminaux de données	● 1 connecteur mâle SUB-D 9 broches ● 1 connecteur femelle SUB-D 25 broches	TSX CTC 07
-	Adaptateur pour équipements terminaux de données	● 1 connecteur mâle SUB-D 9 broches ● 1 connecteur mâle SUB-D 25 broches	TSX CTC 10
-	Adaptateur pour équipement terminal de circuit de données (DCE)	● 1 connecteur femelle SUB-D 9 broches ● 1 connecteur mâle SUB-D 25 broches	TSX CTC 09

NOTE : les câbles et les accessoires de câblage présentés ne sont pas exhaustifs.

Partie III

Mise en œuvre logicielle des communications Modbus Serial et Mode caractère

Dans cette partie

Cette partie présente la mise en œuvre logicielle des communications Modbus Serial et Mode caractère avec le logiciel Unity Pro.

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
4	Méthodologie de mise en œuvre	67
5	Communication Modbus Serial des processeurs BMX P34 1000/2000/2010/20102/2020	71
6	Communication en mode caractère des processeurs BMX P34 1000/2000/2010/20102/2020	105
7	Communications Modbus Serial du BMX NOM 0200	131
8	Communication en Mode caractère du BMX NOM 0200	179
9	Diagnostics de module BMX NOM 0200	215
10	Objets langage des communications Modbus et Mode caractère	221
11	Changement dynamique de protocoles	259

Chapitre 4

Méthodologie de mise en œuvre

Introduction à la phase d'installation

Introduction

L'installation logicielle des modules spécifiques à l'application s'effectue à partir des différents éditeurs d'Unity Pro :

- En mode local
- En mode connecté

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

Phases de mise en œuvre à l'aide d'un processeur

Le tableau suivant présente les différentes phases de l'installation à l'aide d'un processeur :

Phase	Description	Mode
Configuration du processeur	Déclaration du processeur	Hors ligne
	Configuration du port série du processeur	
Configuration du module (le cas échéant)	Déclaration du module	Hors ligne
	Configuration des voies des modules	
	Saisie des paramètres de configuration	
Déclaration des variables	Déclaration des variables de type IODDT spécifiques au processeur/module et des variables du projet	Local (1)
Association	Association des variables IODDT aux voies configurées (éditeur de variables)	Local (1)
Programmation	Programmation du projet.	Local (1)
Génération	Génération (analyse et édition de liens) du projet.	Hors ligne
Transférer	Transfert d'un projet vers l'automate	En ligne
Mise au point	Mise au point du projet depuis les écrans de mise au point et les tables d'animation	En ligne
Documentation	Création d'un fichier de documentation et impression des diverses informations relatives au projet	En ligne
Fonctionnement	Affichage des diverses informations nécessaires à la supervision du projet	En ligne
Légende : (1) Ces différentes phases peuvent aussi être réalisées en mode connecté		

Phases de mise en œuvre à l'aide du simulateur

Le tableau suivant présente les différentes phases de l'installation à l'aide du simulateur :

Phase	Description	Mode
Configuration du processeur	Déclaration du processeur	Hors ligne
	Configuration du port série du processeur	
Configuration du module (le cas échéant)	Déclaration du module	Hors ligne
	Configuration des voies des modules	
	Saisie des paramètres de configuration	
Déclaration des variables	Déclaration des variables de type IODDT spécifiques au processeur/module et des variables du projet	Local (1)
Association	Association des variables IODDT aux voies configurées (éditeur de variables)	Local (1)

Phase	Description	Mode
Programmation	Programmation du projet.	Local (1)
Génération	Génération (analyse et édition de liens) du projet.	Hors ligne
Transférer	Transfert du projet vers le simulateur	En ligne
Simulation	Simulation du programme sans les entrées/sorties	En ligne
Réglage/ Mise au point	Mise au point du projet à partir de tables d'animation.	En ligne
	Modification du programme et des paramètres d'ajustement	
Légende :		
(1) Ces différentes phases peuvent aussi être réalisées en mode connecté		

Configuration du processeur et du module

Les paramètres de configuration sont accessibles uniquement depuis le logiciel Unity Pro.

Création de documentation technique

Unity Pro permet de créer une **documentation technique de projet** (voir *Unity Pro, Modes de marche*).

Le format général de l'impression est constitué de :

- Un titre : référence du module et sa position,
- Une section contenant l'identification du module,
- Une section par voie avec tous les paramètres d'une voie.

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

Chapitre 5

Communication Modbus Serial des processeurs BMX P34 1000/2000/2010/20102/2020

Objet de ce chapitre

Ce chapitre présente la mise en œuvre logicielle de la communication Modbus des processeurs BMX P34 1000/2000/2010/20102/2020.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
5.1	Généralités	72
5.2	Configuration d'une communication Modbus Serial	79
5.3	Programmation d'une communication Modbus Serial	91
5.4	Mise au point d'une communication Modbus Serial	102

Sous-chapitre 5.1

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 Serial	73
Performances	74
Comment accéder aux paramètres de la liaison série	76

A propos de Modbus Serial

Introduction

La communication par Modbus autorise l'échange de données entre tous les appareils connectés au bus. Modbus Serial est un protocole créant 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 : notez que deux maîtres (sur le même bus) n'envoient pas de requêtes simultanément ; sinon, les requêtes seraient perdues et chaque rapport obtiendrait un résultat erroné qui pourrait être 16#0100 (impossible de traiter la requête) ou 16#ODFF (esclave non présent).

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 permettent d'évaluer le temps d'échange typique de la communication Modbus selon différents critères.

Les résultats présentés correspondent à une durée moyenne d'exécution de la fonction READ_VAR en millisecondes.

Définition du temps d'échange

Le temps d'échange est le délai écoulé entre la création d'un échange et la fin de ce dernier. Il inclut la durée de la communication sur la liaison série.

La création de l'échange se produit lors de l'appel de la fonction de communication.

La fin de l'échange se fait sur 1 des événements suivants :

- réception de données,
- Une anomalie se produit.
- temporisation écoulée.

Temps d'échange pour 1 mot

Le tableau suivant présente les temps d'échange d'une communication Modbus d'un processeur BMX P34 2020 pour 1 mot :

Débit de la communication en bits/s	Temps de cycle en ms	Temps d'échange en ms L'esclave Modbus est un BMX P34 1000 cyclique
4800	Cyclique	68
4800	10	72
4800	50	100
9600	Cyclique	35
9600	10	40
9600	50	50
19200	Cyclique	20
19200	10	27
19200	50	50
38400	Cyclique	13
38400	10	20
38400	50	50

Les temps d'échange sont similaires sur les processeurs BMX P34 2020 et BMX P34 2000/2010/20102, et sont inférieurs de 10 % sur le BMX P34 1000.

Temps d'échange pour 100 mots

Le tableau suivant présente les temps d'échange d'une communication Modbus d'un processeur BMX P34 2020 pour 100 mots :

Débit de la communication en bits/s	Temps de cycle en ms	Temps d'échange en ms L'esclave Modbus est un BMX P34 1000 cyclique
4800	Cyclique	500
4800	10	540
4800	50	595
9600	Cyclique	280
9600	10	288
9600	50	300
19200	Cyclique	142
19200	10	149
19200	50	150
38400	Cyclique	76
38400	10	80
38400	50	100

Les temps d'échange sont similaires sur les processeurs BMX P34 2020 et BMX P34 2000/2010/20102, et sont inférieurs de 10 % sur le BMX P34 1000.

Précision de mesure

Tous les temps d'échange cités ci-dessus sont issus de mesures dont la précision est de +/-10 ms.

Comment accéder aux paramètres de la liaison série

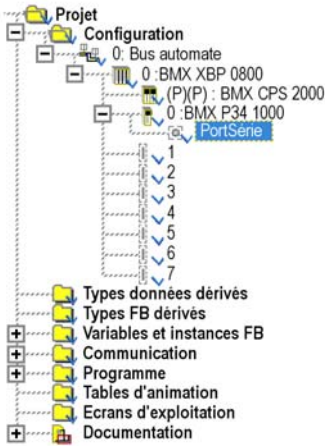
Présentation

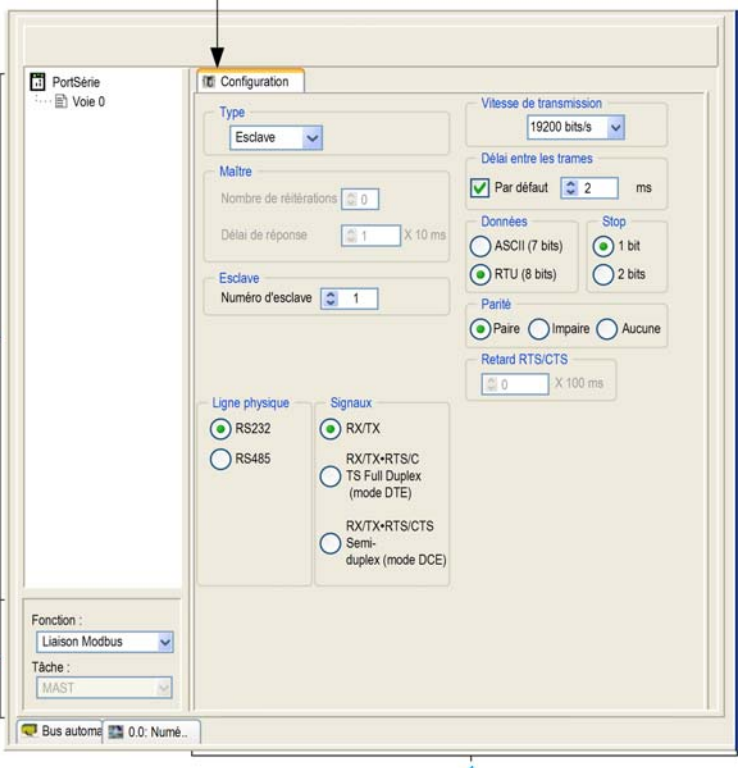
Les pages suivantes expliquent comment accéder à l'écran de configuration du port série pour les processeurs suivants. Vous y trouverez également des informations générales sur les écrans de configuration et de mise au point des liaisons Modbus et de mode caractère :

- 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	A partir du navigateur de projet, déployez le répertoire <i>Station\Configuration\0: PLC bus\0: référence du rack\0: référence du processeur\PortSérie</i>. Résultat : L'écran suivant apparaît : 

Etape	Action
2	Double-cliquez sur le sous-répertoire PortSérie. Résultat : l'écran de configuration apparaît : 

Description de l'écran de configuration

Le tableau suivant présente les différents éléments des écrans de configuration :

Adresse	Élément	Fonction
1	Languettes	L'onglet en avant-plan indique le mode en cours. Chaque mode peut être sélectionné par l'onglet correspondant. Les modes disponibles sont : ● Configuration ● Mise au point (accessible seulement en mode connecté)
2	Zone de voie	Permet : ● de choisir entre le port série et la voie 0 en cliquant sur l'un ou l'autre, ● d'afficher, en cliquant sur le port série, les onglets : ○ "Description" qui donne les caractéristiques de l'équipement, ○ "Objets d'E/S" (voir <i>Unity Pro, Modes de marche</i>) qui permet de présymboliser les objets d'entrées/sorties, ● d'afficher, en cliquant sur la voie, les onglets : ○ Configuration ○ Mise au point ● d'afficher le symbole et le nom de la voie définis par l'utilisateur au travers de l'éditeur de variables.
3	Zone des paramètres généraux	Permet de choisir les paramètres généraux associés à la voie : ● Fonction : les fonctions disponibles sont Modbus ou Mode caractères. Par défaut, la fonction Modbus est configurée. ● Tâche : définit la tâche maître dans laquelle seront échangés les objets à échange implicite de la voie. Cette zone apparaît en grisé et n'est donc pas configurable.
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 5.2

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 d'une communication Modbus Serial	80
Fonctions Modbus accessibles	82
Valeurs par défaut des paramètres d'une communication Modbus Serial	83
Paramètres Modbus liés à l'application	84
Paramètres Modbus liés à la transmission	86
Paramètres liés aux signaux et à la ligne physique dans Modbus	88

Ecran de configuration d'une communication Modbus Serial

Général

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

Accès à l'écran de configuration

Pour accéder à l'écran de configuration d'une communication Modbus Serial, ouvrez le répertoire PortSérie dans le navigateur de projet (*voir page 76*).

Ecran de configuration d'une communication Modbus Serial

La figure ci-dessous représente un écran de configuration par défaut d'une communication Modbus Serial :

The screenshot shows a software interface for configuring a Modbus Serial communication. The window has a title bar with 'PortSérie' and 'Voie 0'. The main area is divided into several sections. On the left, there's a tree view showing 'PortSérie' and 'Voie 0'. The main panel is titled 'Configuration'. It contains the following settings:

- Type:** A dropdown menu set to 'Esclave'.
- Maître:** A section with 'Nombre de répétitions' set to 0 and 'Délai de réponse' set to 1 x 10 ms.
- Esclave:** A section with 'Numéro d'esclave' set to 1.
- Vitesse de transmission:** A dropdown menu set to 19200 bits/s.
- Délai entre les trames:** A section with a checked 'Par défaut' checkbox and a value of 2 ms.
- Données:** A section with 'ASCII (7 bits)' and 'RTU (8 bits)' radio buttons, and 'Stop' settings for 1 bit and 2 bits.
- Parité:** A section with 'Paire', 'Impaire', and 'Aucune' radio buttons.
- Retard RTS/CTS:** A section with a value of 0 x 100 ms.
- Ligne physique:** A section with 'RS232' and 'RS485' radio buttons.
- Signaux:** A section with 'RX/TX', 'RX/TX-RTS/C TS Full Duplex (mode DTE)', and 'RX/TX-RTS/CTS Semi-duplex (mode DCE)' radio buttons.

At the bottom of the window, there are two dropdown menus: 'Fonction' set to 'Liaison Modbus' and 'Tâche' set to 'MAST'. The status bar at the very bottom shows 'Bus automate' and '0.0: Numé...'.

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 certains paramètres peuvent ne pas être accessibles et apparaissent en grisé.

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

Touche	Élément	Commentaire
1	les paramètres concernant l'application, <i>(voir page 84)</i>	Ces paramètres sont accessibles à partir de 3 zones : ● Type, ● Maître, ● Esclave.
2	les paramètres concernant la transmission, <i>(voir page 86)</i>	Ces paramètres sont accessibles à partir de cinq zones : ● Vitesse de transmission, ● Délai entre les trames, ● Données, ● Bits d'arrêt, ● Parité.
3	les paramètres concernant les signaux et la ligne physique. <i>(voir page 88)</i>	Ces paramètres sont accessibles à partir de 3 zones : ● Ligne physique, ● Signaux, ● Retard RTS/CTS.

NOTE : Pour la configuration de la communication Modbus Serial en mode maître, la zone esclave est grisée et ne peut pas être modifiée, et vice-versa.

Fonctions Modbus accessibles

Présentation

L'accessibilité de la fonction de configuration de la liaison série pour les processeurs qui utilisent la liaison série Modbus dépend de la liaison physique adoptée :

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

Fonctions accessibles

Le tableau ci-dessous indique les différentes fonctions configurables selon le type de liaison série utilisée :

Fonction	Liaison RS 485	Liaison RS 232
Nombre de réitérations du maître	X	X
Délai de réponse du maître	X	X
Numéro d'esclave	X	X
Vitesse de transmission	X	X
Délai entre les trames	X	X
Données	● ASCII (7 bits) ● RTU (8 bits)	● ASCII (7 bits) ● RTU (8 bits)
Arrêter	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Parité	● Impaire ● Paire ● Rien	● Impaire ● Paire ● Rien
Signaux RX/TX	X	X
Signaux RTS/CTS	-	X
Retard RTS/CTS	-	X

- X Fonction accessible
- Fonction inaccessible

Valeurs par défaut des paramètres d'une communication Modbus Serial

Présentation

Les paramètres d'une communication Modbus Serial possèdent tous des valeurs par défaut.

Valeurs par défaut

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

Paramètre de configuration	Valeur
Mode	Esclave
Tracé physique	RS232
Numéro d'esclave	1
Délai entre les trames	2 ms
Vitesse de transmission	19200 bits/s
Parité	Paire
Bits de données	RTU (8 bits)
Bits d'arrêt	1 bit

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 :

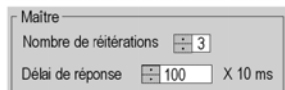


Cette zone vous permet de sélectionner le type de Modbus Serial à utiliser :

- **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 :

A screenshot of a software configuration window titled "Esclave". Inside the window, there is a label "Numéro d'esclave" followed by a text input field containing the number "7". The window has a standard grey border and a title bar.

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 Modbus liés à la transmission

Présentation

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

Ces paramètres sont accessibles à partir de 5 zones :

- Zone Vitesse de transmission,
- Zone Délai inter-caractères,
- Zone Données,
- Zone Stop,
- Zone Parité.

Zone Vitesse de transmission

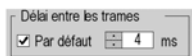
Cette zone de configuration apparaît dans l'écran comme représenté ci-dessous :



Cette zone permet de sélectionner la vitesse de transmission de Modbus Serial. Cette vitesse est en conformité avec les autres équipements. Les valeurs configurables sont 300, 600, 1200, 2400, 4800, 9600, 19200 et 38400 bits/s.

Zone Délai entre les trames

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



Le délai entre les trames est le délai minimal séparant 2 trames en réception. Ce délai est géré quand l'automate est en réception de messages, qu'il soit maître ou esclave.

NOTE : la valeur par défaut dépend de la vitesse de transmission choisie.

NOTE : le délai entre les trames doit correspondre à la valeur par défaut pour être conforme au protocole Modbus. Si un esclave n'est pas conforme, la valeur peut être modifiée et doit être identique pour le maître et tous les esclaves du bus.

Zone Données

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



Cette zone permet de renseigner le type de codage utilisé pour communiquer avec Modbus Serial. Ce champ est à configurer en fonction des autres équipements connectés sur le bus. Deux modes sont configurables :

- Mode RTU :
 - les caractères sont codés sur 8 bits,
 - la fin de trame est détectée par un silence d'au moins 3,5 caractères,
 - l'intégrité de la trame est contrôlée à l'aide d'un mot appelé checksum CRC contenu dans celle-ci.
- Mode ASCII :
 - les caractères sont codés sur 7 bits,
 - le début de trame est détecté par réception du caractère ":",
 - la fin de trame est détectée par un retour chariot et un saut de ligne,
 - l'intégrité de la trame est contrôlée à l'aide d'un octet appelé checksum LRC contenu dans celle-ci.

Zone Stop

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



Cette zone permet de renseigner le nombre de bits d'arrêt utilisés pour communiquer. Ce champ est à configurer en fonction des autres équipements. Les valeurs configurables sont :

- 1 bit
- 2 bits

Zone Parité

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



Cette zone permet de définir l'adjonction ou non d'un bit de parité ainsi que son type. Ce champ est à configurer en fonction des autres équipements. Les valeurs configurables sont :

- Paire
- Impaire
- Rien

Paramètres liés aux signaux et à la ligne physique dans Modbus

Présentation

Après avoir configuré la voie de communication, vous devez renseigner les paramètres de la ligne physique et des signaux.

Ces paramètres sont accessibles à partir de 3 zones :

- Zone Ligne Physique,
- Zone Signaux,
- Zone Retard RTS/CTS.

Zone Ligne Physique

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

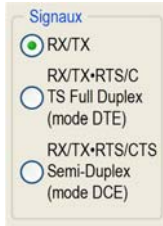


Cette zone permet de choisir le type de la ligne physique du port série des processeurs BMX P34 1000/2000/2010/20102/2020 parmi les 2 types suivants :

- ligne RS 232,
- ligne RS 485.

Zone Signaux

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



Cette zone permet de sélectionner les signaux supportés par la ligne physique RS 232 :

- RX/TX
- RX/TX + RTS/CTS (mode DTE)
- RX/TX + RTS/CTS (mode DCE)

Si la ligne configurée est de type RS 485, la zone est entièrement grisée et la valeur par défaut est RX/TX.

NOTE : seuls les signaux RX/TX et RX/TX + RTS/CTS sont disponibles dans la configuration du port série des processeurs BMX P34 1000/2000/2010/20102/2020.

Zone retard RTS/CTS

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



La zone de temporisation RTS/CTS n'est disponible que quand les deux cases RS232 et RX/TX+RTS/CTS sont cochées. Un algorithme de contrôle de flux RTS/CTS est sélectionné si la valeur par défaut est de 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) par les points suivants :

- L'algorithme de contrôle de flux RTS/CTS est associé au débordement du tampon de réception (full duplex).
- L'algorithme de contrôle de modem RTS/CTS utilise une procédure de communication partagée, par exemple sur un modem radio.

Algorithme de contrôle de flux RTS/CTS

L'objectif est d'éviter un débordement du tampon de réception.

Le signal de sortie RTS de chaque appareil est relié au signal d'entrée CTS de l'autre appareil. L'émetteur (M340) est autorisé à envoyer des données lorsqu'il reçoit le signal d'entrée RTS (par exemple d'un autre M340) sur son entrée CTS. Cet algorithme est symétrique et autorise une communication asynchrone full duplex.

Algorithme de contrôle de modem RTS/CTS

Avant envoi d'une requête, l'émetteur (M340) active le signal RTS et attend le déclenchement du signal CTS par le modem. Si CTS n'est pas activé après la temporisation RTS/CTS, la requête est ignorée.

Sous-chapitre 5.3

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	92
Services pris en charge par le processeur esclave d'une liaison Modbus	100

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

Présentation

Lorsqu'ils sont utilisés comme processeur maître dans une liaison Modbus, les processeurs suivants prennent en charge plusieurs services à l'aide des fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH.

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

Echanges de données

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

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

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

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 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.

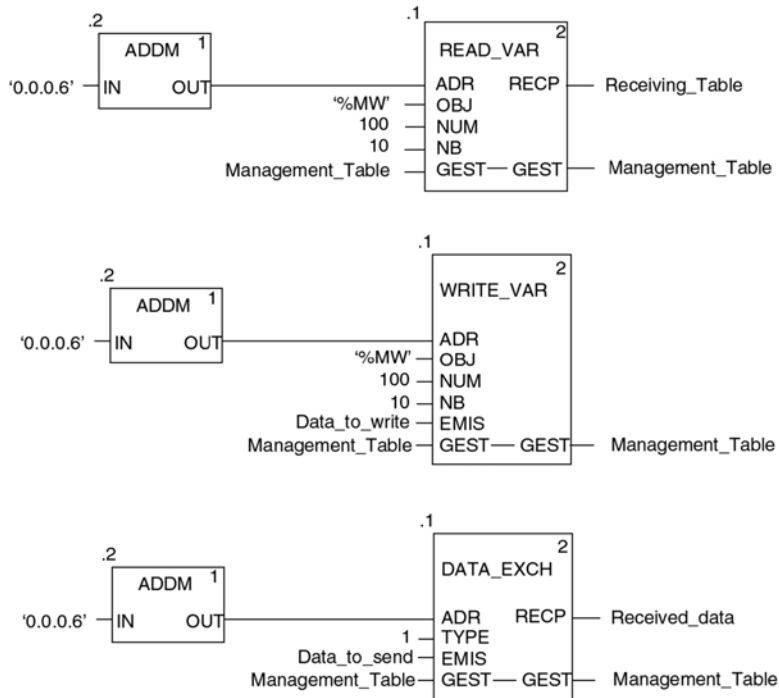
Fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH

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

- READ_VAR : lecture de variables.
- WRITE_VAR : écriture de variables.
- DATA_EXCH : Pour envoyer des requêtes Modbus à un autre appareil par le protocole sélectionné

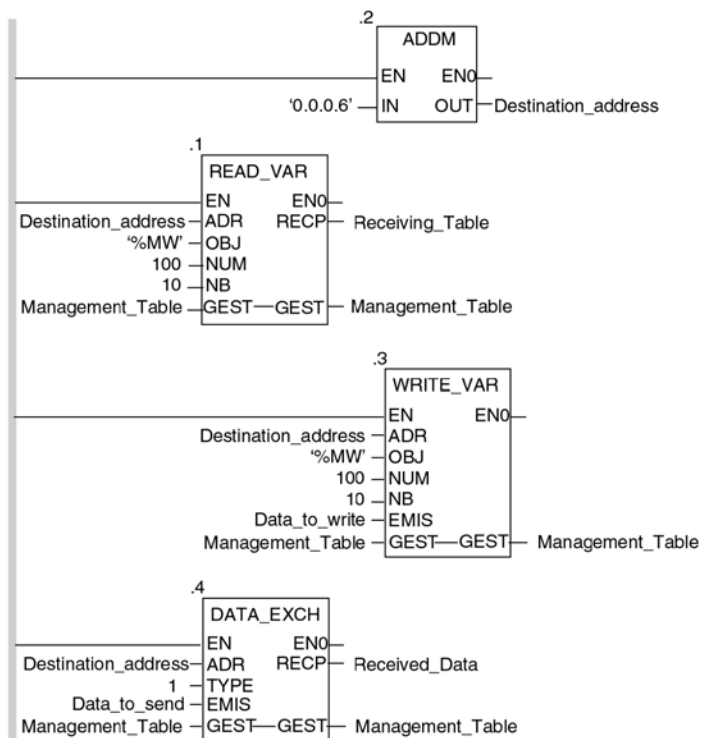
Exemple de programmation en FBD

Le schéma ci-dessous représente un exemple de programmation des fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH dans le langage FBD :



Exemple de programmation en Ladder

La figure suivante représente un exemple de programmation en langage Ladder des fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH :



Exemple de programmation en ST

Les lignes de code ci-dessous représentent un exemple de programmation des fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH dans le langage ST :

```
READ_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Management_Table,  
Receiving_Table);
```

```
WRITE_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Data_to_write,  
Management_Table);
```

```
DATA_EXCH(ADDM('0.0.0.6'), 1, Data_to_send, Management_Table,  
Received_data);
```

Annulation d'un échange

Un échange exécuté par les fonctions READ_VAR, WRITE_VAR et DATA_EXCH peut être annulé par l'une ou l'autre méthode de programmation, toutes les deux présentées en langage ST ci-dessous:

- Utilisation de la fonction CANCEL :

```
IF (%MW40.0) THEN
  %MW200:=SHR(%MW40,8);
  CANCEL(%MW200,%MW185);
END_IF;
```

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction READ_VAR et est à 1 lorsque la fonction de communication est active. Si ce bit est à 1, le programme exécute les instructions suivantes :

- décale à droite les bits de %MW40 d'un octet (8 bits) et charge l'octet correspondant au numéro d'échange de la communication dans le mot %MW200,
- annule l'échange dont le numéro est contenu dans le mot %MW200 à l'aide de la fonction CANCEL.

- utilisation du bit d'annulation de la fonction de communication :

```
IF (%MW40.0) THEN
  SET(%MW40.1);
  READ_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, %MW40:4, %MW10:10);
END_IF;
```

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction READ_VAR et est à 1 lorsque la fonction de communication est active. Si ce bit est défini sur 1, le programme met à 1 le bit %MW40.1, qui correspond au bit d'annulation de la fonction. Ceci a pour effet d'arrêter la communication de la fonction READ_VAR.

NOTE : lors de l'utilisation du bit d'annulation de la fonction de communication contenu dans le mot de gestion d'échanges de la fonction (%MW40 dans cet exemple), la fonction (READ_VAR dans cet exemple) doit être appelée dans le but d'activer l'annulation de l'échange.

NOTE : Lors de l'utilisation du bit d'annulation de la fonction de communication, il est possible d'annuler une communication à partir d'une table d'animation. Pour cela, il suffit de mettre à 1 le bit d'annulation de la fonction (%MW40.1 dans cet exemple), puis de redémarrer la fonction de communication.

NOTE : Cet exemple de programmation concerne la fonction READ_VAR, mais il est tout aussi applicable aux fonctions WRITE_VAR ainsi que DATA_EXCH.

NOTE : l'utilisation de la fonction CANCEL utilise un mot de compte rendu pour la fonction CANCEL (%MW185 pour cet exemple).

Description des paramètres de la fonction ADDM

Le tableau suivant décrit les différents paramètres de la fonction ADDM :

Paramètre	Type	Description
IN	STRING	Adresse d'un équipement sur un bus ou sur une liaison série. La syntaxe de l'adresse est du type "r.m.c.node". L'adresse est composée des paramètres suivants : ● r : numéro du rack du processeur toujours égal à 0, ● m : emplacement du processeur dans le rack toujours égal à 0, ● c : numéro de voie toujours égal à 0 car la liaison série d'un processeur est toujours la voie 0, ● node : numéro de l'esclave destinataire de la requête.
OUT	ARRAY [0..7] OF INT	Tableau représentant l'adresse d'un équipement. Ce paramètre est utilisable par plusieurs fonctions de communication comme paramètre d'entrée.

Description des paramètres de la fonction READ_VAR

Le tableau suivant décrit les différents paramètres de la fonction READ_VAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse de l'entité destinataire donnée par le paramètre OUT de la fonction ADDM.
OBJ	STRING	Type d'objet à lire. Les types disponibles sont les suivants : ● %M : bit interne, ● %MW : mot interne, ● %I : bit d'entrée externe, ● %IW : mot d'entrée externe.
NUM	DINT	Adresse du premier objet à lire.
NB	INT	Nombre d'objets consécutifs à lire.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● mot de rang 1 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Numéro d'échange, ○ Octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1). ● mot de rang 2 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Compte rendu d'opération, ○ Octet de poids faible : Compte rendu de communication. ● mot de rang 3 : mot géré par l'utilisateur et qui définit le temps d'attente maximal de réponse sur une base de 100 ms. ● mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.
RECP	ARRAY [n..m] OF INT	Tableau de mots contenant la valeur des objets lus.

Description des paramètres de la fonction WRITE_VAR

Le tableau suivant décrit les différents paramètres de la fonction WRITE_VAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse de l'entité destinataire donnée par le paramètre OUT de la fonction ADDM.
OBJ	STRING	Type d'objet à écrire. Les types disponibles sont les suivants : ● %M : bit interne, ● %MW : mot interne, Remarque : WRITE_VAR ne peut pas être utilisée pour les variables %I et %IW.
NUM	DINT	Adresse du premier objet à écrire.
NB	INT	Nombre d'objets consécutifs à écrire.
EMIS	ARRAY [n..m] OF INT	Tableau de mots contenant la valeur des objets à écrire.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● mot de rang 1 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Numéro d'échange, ○ Octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1). ● mot de rang 2 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Compte rendu d'opération, ○ Octet de poids faible : Compte rendu de communication. ● mot de rang 3 : mot géré par l'utilisateur et qui définit le temps d'attente maximal de réponse sur une base de 100 ms. ● mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.

Description des paramètres de la fonction DATA_EXCH

Le tableau suivant décrit les différents paramètres de la fonction DATA_EXCH :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse de l'entité destinataire donnée par le paramètre OUT de la fonction ADDM.
TYPE	INT	Pour les automates Modicon M340, la seule valeur possible est : 1 : Envoi d'un tableau EMIS, l'automate attend ensuite la réception d'un tableau RECP.
EMIS	ARRAY [n..m] OF INT	Tableau d'entiers à envoyer à l'appareil destination de la requête. Remarque : Il est impératif que la longueur des données à envoyer (en octets) soit affectée au quatrième mot du tableau de gestion avant le lancement de la fonction, pour que celle-ci puisse être exécutée correctement.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● mot de rang 1 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Numéro d'échange, ○ Octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1). ● mot de rang 2 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Compte rendu d'opération, ○ Octet de poids faible : Compte rendu de communication. ● mot de rang 3 : mot géré par l'utilisateur et qui définit le temps d'attente maximal de réponse sur une base de 100 ms. ● mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.
RECP	ARRAY [n..m] OF INT	Tableau d'entiers contenant les données reçues. Remarque : La taille des données reçues (en octets) est automatiquement écrite par le système dans le quatrième mot du tableau de gestion.

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

Désignation	Code fonction / Code sous-fonction
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 5.4

Mise au point d'une communication Modbus Serial

Ecran de mise au point d'une communication Modbus Serial

Général

L'écran de mise au point d'une communication Modbus Serial 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 Serial :

Etape	Action
1	Accédez à l'écran de configuration d'une communication Modbus Serial. (voir page 80)
2	Sélectionnez l'onglet "Mise au point" dans l'écran qui apparaît.

Description de l'écran de mise au point

L'écran de mise au point se décompose en 2 zones :

- la zone Type,
- la zone Compteurs.

Zone Type

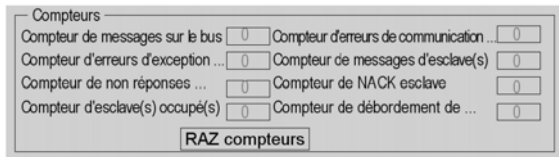
La zone se présente comme ci-dessous :



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

Zone Compteurs

La zone se présente comme ci-dessous :



Cette zone affiche les différents compteurs de mise au point.

Le bouton R.A.Z. 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 sont :

- Compteur de messages sur le bus : 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 : 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'esclave(s) : nombre d'erreurs d'exception Modbus détectées par le processeur.
- Compteur de messages d'esclave(s) : nombre de message reçus et traités par la liaison Modbus,
- Compteur de non réponses d'esclave(s) : nombre de messages émis par le système distant pour lesquels il ne retourne pas de réponse (ni une réponse normale ni une réponse d'exception). Ce compteur compte aussi le nombre de messages reçus en mode diffusion.
- Compteur d'accusés de réception négatifs d'esclave(s) : nombre de messages adressés au système distant pour lesquels il retourne un accusé de réception négatif,
- Compteur d'esclave(s) occupé(s) : nombre de messages adressés au système distant pour lesquels il retourne une réponse d'exception esclave occupé,
- Compteur de dépassements de caractères sur le bus : nombre de messages adressés au processeur qu'il ne peut pas acquérir à cause d'un dépassement de caractères. Un 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 6

Communication en mode caractère des processeurs BMX P34 1000/2000/2010/20102/2020

Objet de cette section

Ce chapitre vous présente la mise en œuvre logicielle de la communication en Mode caractère des processeurs BMX P34 1000/2000/2010/20102/2020.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
6.1	Généralités	106
6.2	Configuration d'une communication en mode caractère	110
6.3	Programmation d'une communication en mode caractère	122
6.4	Mise au point d'une communication en mode caractère	128

Sous-chapitre 6.1

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 en mode caractère	107
Performances	108

A propos de la communication en mode caractère

Introduction

La communication en Mode caractère permet de réaliser des fonctions de dialogue et de communication entre les automates et des équipements parmi les systèmes suivants :

- périphériques usuels (imprimante, écran-clavier, terminal d'atelier,...),
- périphériques spécialisés (lecteur de codes barres par exemple),
- calculateurs (calculateur de supervision, de gestion de production,...),
- équipements hétérogènes (commandes numériques, variateurs de vitesse,...),
- 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 des temps d'échange types en mode caractère.

Les résultats présentés correspondent à une durée moyenne d'exécution de la fonction PRINT_CHAR en millisecondes.

Définition du temps d'échange

Le temps d'échange est le délai écoulé entre la création d'un échange et la fin de ce dernier. Il inclut la durée de la communication sur la liaison série.

La création de l'échange se produit lors de l'appel de la fonction de communication.

La fin de l'échange se fait sur 1 des événements suivants :

- réception des données,
- une anomalie,
- expiration du timeout.

Temps d'échanges pour 80 caractères

Le tableau ci-dessous indique les temps d'échange pour la transmission de 80 caractères en mode caractère sur un processeur BMX P34 2020 :

Débit de la communication en bits/s	Temps de cycle en ms	Temps d'échange en ms
1200	10	805
1200	20	820
1200	50	850
1200	100	900
1200	255	980
4800	10	210
4800	20	220
4800	50	250
4800	100	300
4800	255	425
9600	10	110
9600	20	115
9600	50	145
9600	100	200
9600	255	305
19200	10	55

19200	20	60
19200	50	95
19200	100	100
19200	255	250

Les temps d'échange des processeurs BMX P34 2000/2010/20102 sont similaires à ceux du BMX P34 2020. Les temps d'échange du BMX P34 1000 sont inférieurs de 10 %.

Précision de mesure

Tous les temps d'échange cités ci-dessus sont issus de mesures dont la précision est de +/-10 ms.

Sous-chapitre 6.2

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 en mode caractère	111
Fonctions accessibles en mode caractère	114
Valeurs par défaut des paramètres d'une communication en mode caractère	115
Paramètre de détection de fin de message en mode caractère	116
Paramètres liés à la transmission en mode caractère	118
Paramètres liés aux signaux et à la ligne physique en mode caractère	120

Ecran de configuration d'une communication en mode caractère

Général

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

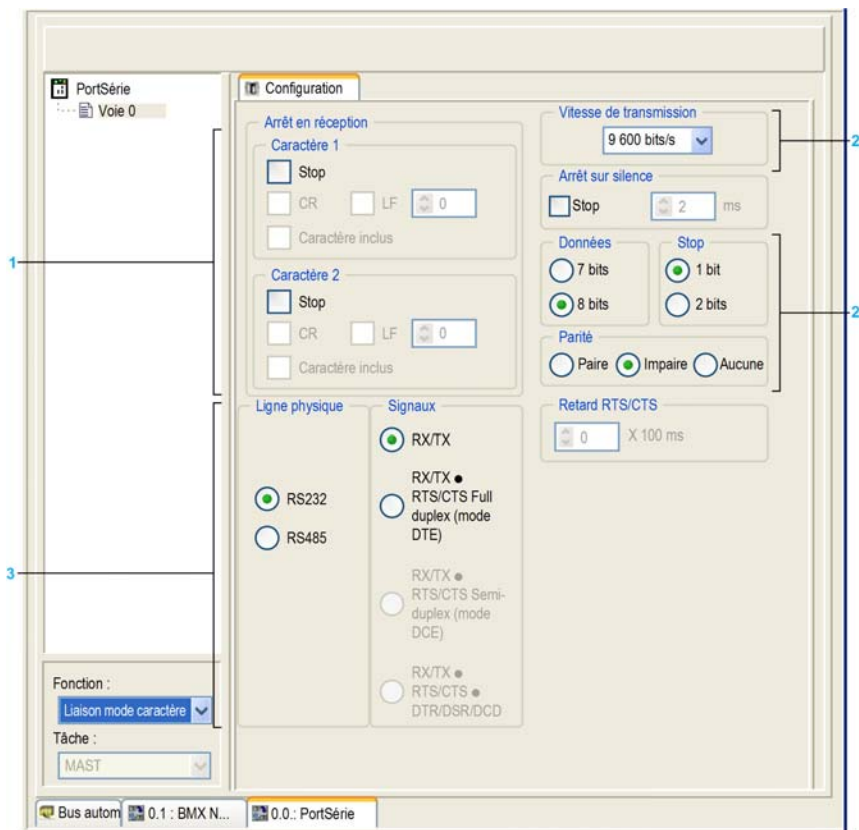
Accès à l'écran de configuration

Le tableau suivant présente 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 de projet (<i>voir page 76</i>).
2	Sélectionnez la fonction Liaison mode caractères dans l'écran qui apparaît.

Ecran de configuration d'une communication en mode caractère

La figure ci-dessous représente l'écran de configuration par défaut d'une communication en mode caractère :



Description

Ces zones permettent de configurer les paramètres de voie. En mode connecté, ces zones sont accessibles. En mode local, elles sont accessibles, mais certains paramètres ne le sont pas. Ils sont alors grisés.

Le tableau ci-dessous présente les différentes zones de l'écran de configuration de liaison de communication en mode caractère :

Touche	Élément	Commentaire
1	les paramètres concernant la détection de fin de message, (voir page 116)	Ces paramètres sont accessibles à partir de 2 zones : ● Arrêt en réception, ● Arrêt sur silence.
2	les paramètres concernant la transmission, (voir page 118)	Ces paramètres sont accessibles à partir de 4 zones : ● Vitesse de transmission, ● Données, ● Bits d'arrêt, ● Parité.
3	les paramètres concernant les signaux et la ligne physique. (voir page 120)	Ces paramètres sont accessibles à partir de 3 zones : ● Ligne physique, ● Signaux, ● Retard RTS/CTS.

Fonctions accessibles en mode caractère

Présentation

L'accessibilité de la fonction de configuration de la liaison série pour les processeurs qui utilisent le protocole mode de caractère dépend de la liaison physique adoptée :

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

Fonctions accessibles

Le tableau ci-dessous indique les différentes fonctions configurables selon le type de liaison série utilisée :

Fonction	Liaison RS 485	Liaison RS 232
Vitesse de transmission	X	X
Données	● 7 bits ● 8 bits	● 7 bits ● 8 bits
Arrêter	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Parité	● Impaire ● Paire ● Rien	● Impaire ● Paire ● Rien
Arrêt en réception	X	X
Arrêt sur silence	X	X
Signaux RX/TX	X	X
Signaux RTS/CTS	-	X
Retard RTS/CTS	-	X

- X Fonction accessible
- Fonction inaccessible

Valeurs par défaut des paramètres d'une communication en mode caractère

Présentation

Les paramètres d'une communication en Mode caractère possèdent tous des valeurs par défaut.

Valeurs par défaut

Le tableau suivant présente les valeurs par défaut des paramètres d'une communication en Mode caractère :

Paramètre de configuration	Valeur
Ligne physique	RS 232
Vitesse de transmission	9 600 bits/s
Parité	Impaire
Bits de données	8 bits
Bits d'arrêt	1 bit

Paramètre de détection de fin de message en mode caractère

Présentation

Après avoir configuré la voie de communication, vous devez renseigner les paramètres dédiés aux détections de fins de messages.

Ces paramètres sont accessibles à partir de 2 zones :

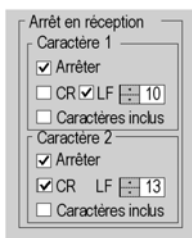
- zone Arrêt en réception : condition d'arrêt par caractère spécial,
- zone Arrêt sur silence : condition d'arrêt par silence.

Conditions d'utilisation

La sélection de l'arrêt sur silence entraîne la désélection de l'arrêt par caractère et inversement.

Zone Arrêt en réception

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



Arrêt en réception

Caractère 1

☒ Arrêter

☐ CR ☒ LF : 10

☐ Caractères inclus

Caractère 2

☒ Arrêter

☒ CR ☐ LF : 13

☐ Caractères inclus

Une requête de réception peut se terminer dès la réception d'un caractère particulier.

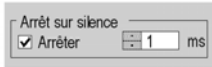
En cochant l'option Arrêt, il est possible d'activer et de paramétrer l'arrêt en réception par un caractère de fin :

- CR : permet de détecter la fin du message par un retour chariot,
- LF : permet de détecter la fin de message par un retour à la ligne,
- champ de saisie : permet d'identifier un caractère de fin, différent du caractère CR ou LF, par une valeur décimale :
 - 0...255 si les données sont codées sur 8 bits,
 - 0...127 si les données sont codées sur 7 bits.
- caractère inclus : permet d'inclure le caractère de fin dans la table de réception de l'application automate.

Il est possible de configurer 2 caractères de fin de réception. Dans la fenêtre ci-dessus, la fin de réception d'un message est détectée par le caractère LF ou CR.

Zone Arrêt sur silence

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

A screenshot of a configuration window titled 'Arrêt sur silence'. It contains a checked checkbox labeled 'Arrêter' and a text input field with the value '1' followed by 'ms'.

Cette zone permet de détecter la fin d'un message en réception par absence de caractère de fin pendant un temps donné.

L'arrêt sur silence est valide lorsque la case Arrêt est cochée. La durée du silence, exprimée en millisecondes, est fixée par le champ de saisie.

NOTE : Les valeurs disponibles sont comprises entre 1 ms et 10 000 ms et dépendent de la vitesse de transmission choisie.

Paramètres liés à la transmission en mode caractère

Présentation

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

Ces paramètres sont accessibles à partir de 4 zones :

- Zone Vitesse de transmission,
- Zone Données,
- Zone Stop,
- Zone Parité.

Zone Vitesse de transmission

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



Cette zone permet de sélectionner la vitesse de transmission du protocole Mode caractère. Cette vitesse est en conformité avec les autres équipements. Les valeurs configurables sont 300, 600, 1 200, 2 400, 4 800, 9 600, 19 200 et 38 400 bits/s.

Zone Données

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



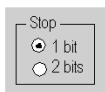
Cette zone permet de renseigner la taille des données échangées sur la liaison. Les valeurs disponibles sont :

- 7 bits
- 8 bits

Il est conseillé d'ajuster le nombre de bits de données en fonction des équipements distants.

Zone Stop

La zone se présente comme ci-dessous :



Cette zone permet de renseigner le nombre de bits d'arrêt utilisés pour communiquer. Il est conseillé d'ajuster le nombre de bits d'arrêt en fonction des équipements distants.

Les valeurs configurables sont :

- 1 bit
- 2 bits

Zone Parité

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



Cette zone permet de définir l'adjonction ou non d'un bit de parité ainsi que son type. Il est conseillé d'ajuster la parité en fonction des équipements distants. Les valeurs configurables sont :

- Paire
- Impaire
- Rien

Paramètres liés aux signaux et à la ligne physique en mode caractère

Présentation

Après avoir configuré la voie de communication, vous devez renseigner les paramètres de la ligne physique et des signaux.

Ces paramètres sont accessibles à partir de 3 zones :

- zone Ligne Physique,
- zone Signaux,
- zone Retard RTS/CTS.

Zone Ligne Physique

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

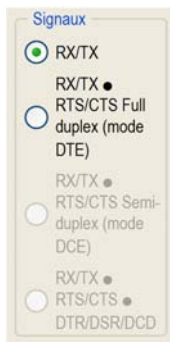


Cette zone permet de choisir la ligne physique du port série des processeurs BMX P34 1000/2000/2010/20102/2020 parmi les 2 types suivants :

- ligne RS 232,
- ligne RS 485,

Zone Signaux

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



Cette zone permet de sélectionner les signaux supportés par la ligne physique RS 232 :

- RX/TX
- RX/TX + RTS/CTS Full Duplex (mode DTE)

Si la ligne configurée est de type RS 485, la zone est entièrement grisée et la valeur par défaut est RX/TX.

NOTE : seuls les signaux RX/TX et RX/TX + RTS/CTS sont disponibles dans la configuration du port série des processeurs BMX P34 1000/2000/2010/20102/2020.

Zone retard RTS/CTS

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



La zone de temporisation RTS/CTS n'est disponible que quand les deux cases RS232 et RX/TX+RTS/CTS sont cochées.

Un algorithme de contrôle de RTS/CTS est sélectionné : Avant l'émission d'une chaîne de caractères, le système attend l'activation du signal CTS (Clear To Send). Cette zone permet de renseigner le temps d'attente maximum entre ces 2 signaux. Si le temps est expiré, la requête n'est pas envoyée sur le bus. Les valeurs configurables sont comprises entre 0 s et 10 s.

NOTE : La valeur par défaut est 0 ms.

NOTE : La valeur 0 s signifie l'absence de gestion du retard entre 2 signaux.

Algorithme de contrôle de flux RTS/CTS

L'objectif est d'éviter un débordement du tampon de réception.

Le signal de sortie RTS de chaque appareil est relié au signal d'entrée CTS de l'autre appareil. L'émetteur (M340) est autorisé à envoyer des données lorsqu'il reçoit le signal d'entrée RTS (par exemple d'un autre M340) sur son entrée CTS. Cet algorithme est symétrique et autorise une communication asynchrone full duplex.

Sous-chapitre 6.3

Programmation d'une communication en mode caractère

Fonctions de communication en 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 (*voir Unity Pro, Communication, Bibliothèque de blocs*) : lecture d'un tableau de 1 024 octets maximum.

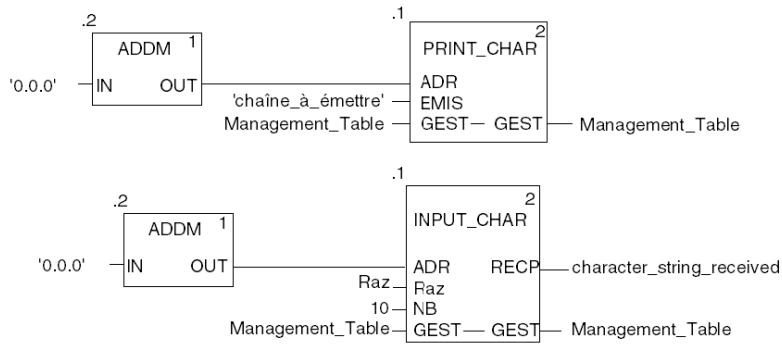
Le port série de l'automate Modicon M340 est en full duplex. Par conséquent, une fonction PRINT_CHAR peut être envoyée même si une fonction INPUT_CHAR est en attente d'envoi.

NOTE : pour acquitter le bit d'activité de la fonction INPUT_CHAR, il est nécessaire de définir une valeur de timeout si la voie est configurée sans arrêt en cas de silence. Pour la fonction PRINT_CHAR, la définition d'un timeout est facultative mais néanmoins conseillée.

NOTE : contrairement au module NOM 0200 sur la liaison RS485, l'UC enregistre l'écho des données émises dans le même tampon que les données reçues. Le tampon de l'UC doit donc être vidé après chaque fonction PRINT_CHAR ou avant l'envoi de données sur 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 exécuter une fonction INPUT_CHAR avec l'option de réinitialisation du tampon activée, puis annuler cette fonction élémentaire avant la fin du timeout.

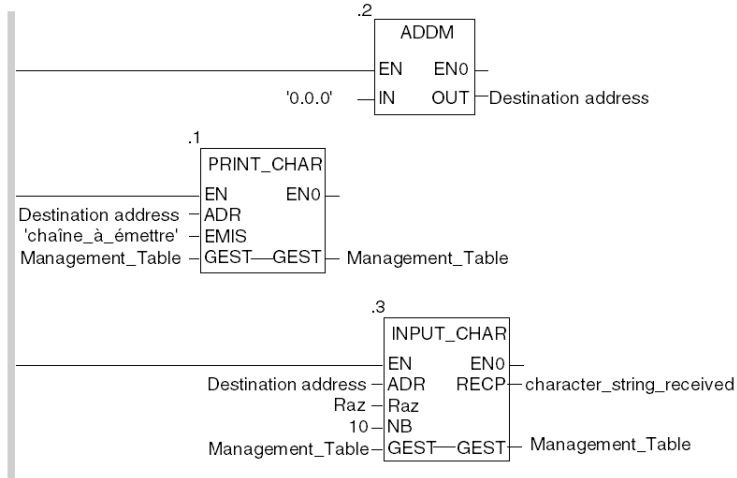
Exemple de programmation en FBD

La figure suivante représente un exemple de programmation en langage FBD des fonctions de communication PRINT_CHAR et INPUT_CHAR :



Exemple de programmation en Ladder

La figure suivante représente un exemple de programmation en langage Schéma à contacts des fonctions de communication PRINT_CHAR et INPUT_CHAR :



Exemple de programmation en ST

Les lignes de code suivantes représentent un exemple de programmation des fonctions de communication PRINT_CHAR et INPUT_CHAR en langage ST :

```
PRINT_CHAR(ADDM('0.0.0'), 'chaîne_à_émettre', Management_Table);  
INPUT_CHAR(ADDM('0.0.0'), reset_integer_to_0, 10, Management_Table,  
character_string_received);
```

Caractéristiques de la fonction INPUT_CHAR

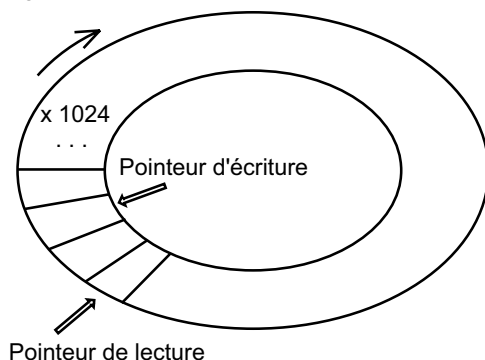
Si le paramètre d'entrée Reset a la valeur 1, le tampon est d'abord réinitialisé, puis le processeur attend la réception de données. L'utilisation de cette fonction est conseillée pour démarrer correctement une réception en éliminant les anciennes données pouvant rester dans le tampon.

Mécanisme interne de l'UC

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

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

La figure ci-dessous représente ce mécanisme :



Annulation d'un échange

Deux types de programmation permettent d'annuler un échange effectué par les fonctions PRINT_CHAR et INPUT_CHAR. Ces deux programmations sont présentées en langage ST ci-dessous :

- Utilisation de la fonction CANCEL :
IF (%MW40.0) THEN
 %MW200:=SHR(%MW40,8);
 CANCEL(%MW200,%MW185);
END_IF;

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction PRINT_CHAR et il est à 1 lorsque la fonction de communication est active. Si ce bit est à 1, le programme exécute les instructions suivantes :

- Il déplace les bits %MW40 d'un octet (8 bits) vers la droite et charge l'octet correspondant au numéro d'échange de la communication dans le mot %MW200.
- Il annule l'échange dont le numéro est contenu dans le mot %MW200 à l'aide de la fonction CANCEL.

- Utilisation du bit d'annulation de la fonction de communication :

```
IF (%MW40.0) THEN
  SET(%MW40.1);
  PRINT_CHAR(ADDM('0.0.0'), 'chaîne_à_émettre', %MW40:4);
END_IF;
```

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction PRINT_CHAR et il est à 1 lorsque la fonction de communication est active. Si ce bit est à 1, le programme met à 1 le bit %MW40.1 d'annulation de la fonction. Ceci a pour effet d'arrêter la communication de la fonction PRINT_CHAR.

NOTE : en cas d'utilisation du bit d'annulation de la fonction de communication, la fonction doit être appelée dans le but d'activer le bit d'annulation contenu dans le mot de gestion d'échanges de la fonction (%MW40 dans l'exemple).

NOTE : Lors de l'utilisation du bit d'annulation de la fonction de communication, il est possible d'annuler une communication à partir d'une table d'animation. Pour cela, il suffit de mettre à 1 le bit d'annulation de la fonction (%MW40.1 dans l'exemple).

NOTE : cet exemple de programmation concerne la fonction PRINT_CHAR, mais il est également applicable à la fonction INPUT_CHAR.

NOTE : la fonction CANCEL utilise un mot de compte rendu (%MW185 dans cet exemple).

Description des paramètres de la fonction ADDM

Le tableau suivant décrit les différents paramètres de la fonction ADDM :

Paramètre	Type	Description
IN	STRING	Adresse d'un équipement sur un bus ou sur une liaison série. La syntaxe de l'adresse est du type "r.m.c.node". L'adresse est composée des paramètres suivants : ● r : numéro du rack du système destinataire, toujours égal à 0 ● m : numéro d'emplacement du système destinataire dans le rack, toujours égal à 0 ● c : numéro de voie, toujours égal à 0 car la liaison série d'un système distant est toujours la voie 0 ● node : champ facultatif pouvant être SYS ou vide
OUT	ARRAY [0..7] OF INT	Tableau représentant l'adresse d'un équipement. Ce paramètre est utilisable par plusieurs fonctions de communication comme paramètre d'entrée.

Description des paramètres de la fonction PRINT_CHAR

Le tableau suivant décrit les différents paramètres de la fonction PRINT_CHAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse du message recevant la voie en mode caractère donnée par le paramètre OUT de la fonction ADDM.
EMIS	STRING	Chaîne de caractères à émettre.
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● Mot de rang 1 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : numéro d'échange, ○ octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1) ● Mot de rang 2 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : compte rendu d'opération ○ octet de poids faible : compte rendu de communication ● Mot de rang 3 : mot géré par l'utilisateur et qui définit le temps de réponse maximum en utilisant une base de temps de 100 ms. ● Mot de rang 4 : mot géré par l'utilisateur et qui définit la longueur de l'échange. ○ Si cette longueur de paramètre est définie sur 0, le système envoie la chaîne entière. ○ Si cette longueur de paramètre est supérieure à la longueur de la chaîne, l'erreur 16#0A (Taille insuffisante du tampon d'envoi) est renvoyée dans le 2e mot de gestion et aucun caractère n'est envoyé. Remarque : pour la fonction PRINT_CHAR, le 4e mot de gestion (longueur à envoyer) ne doit pas être écrit alors que le bit d'activité est à 1 (c'est-à-dire pendant l'exécution de la fonction élémentaire). Sinon, la fonction PRINT_CHAR risquerait de se verrouiller en renvoyant une erreur 0xB.

Description des paramètres de la fonction INPUT_CHAR

Le tableau suivant décrit les différents paramètres de la fonction INPUT_CHAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse du message recevant la voie en mode caractère donnée par le paramètre OUT de la fonction ADDM.
RAZ	INT	Réinitialisation. Ce paramètre permet de réinitialiser la mémoire de réception du coupleur : ● Valeur = 0 : pas de réinitialisation de la mémoire ● Valeur = 1 : réinitialisation de la mémoire
NB	INT	Longueur de la chaîne de caractères à recevoir.
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● Mot de rang 1 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : numéro d'échange, ○ Octet de poids faible : bit d'activité (rang 0), bit d'annulation (rang 1) et bit de reconnaissance immédiate (rang 2) ● Mot de rang 2 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : compte rendu d'opération ○ octet de poids faible : compte rendu de communication ● Mot de rang 3 : mot géré par l'utilisateur et qui définit le temps de réponse maximum en utilisant une base de temps de 100 ms. ● Mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.
RECP	STRING	Chaîne de caractères reçue. Cette chaîne est mémorisée dans une chaîne de caractères.

Sous-chapitre 6.4

Mise au point d'une communication en mode caractère

Ecran de mise au point d'une communication en mode caractère

Général

L'écran de mise au point d'une communication en 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 d'une communication en Mode caractère :

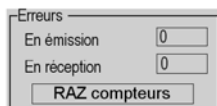
Etape	Action
1	Accédez à l'écran de configuration d'une communication en Mode caractère. (voir page 111)
2	Sélectionnez l'onglet "Mise au point" dans l'écran qui apparaît.

Description de l'écran de 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 :



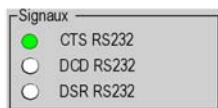
Cette zone indique le nombre d'interruptions de communication comptabilisées par le processeur :

- **en transmission** : correspond au nombre d'interruptions en émission (image du mot %MW4),
- **en réception** : correspond au 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 traité par le processeur (aucune activité pour ce voyant).
- **DSR RS232** : non traité par le processeur (aucune activité pour ce voyant).

Chapitre 7

Communications Modbus Serial du BMX NOM 0200

Objet de ce chapitre

Ce chapitre présente la mise en œuvre logicielle de la communication Modbus Serial du BMX NOM 0200.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
7.1	Généralités	132
7.2	Configuration d'une communication Modbus Serial	139
7.3	Programmation d'une communication Modbus Serial	156
7.4	Mise au point d'une communication Modbus Serial	175

Sous-chapitre 7.1

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 Serial	133
Performances	134
Comment accéder aux paramètres de la liaison série	136

A propos de Modbus Serial

Introduction

La communication par Modbus autorise l'échange de données entre tous les appareils connectés au bus. Modbus Serial est un protocole créant 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 : notez que deux maîtres (sur le même bus) n'envoient pas de requêtes simultanément ; sinon, les requêtes seraient perdues et chaque rapport obtiendrait un résultat erroné qui pourrait être 16#0100 (impossible de traiter la requête) ou 16#ODFF (esclave non présent).

AVERTISSEMENT

PERTE DE DONNEES CRITIQUES

N'utilisez le port 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 permettent d'évaluer le temps d'échange typique de la communication Modbus selon différents critères.

Les résultats présentés correspondent à une durée moyenne d'exécution de la fonction READ_VAR en millisecondes.

Définition du temps d'échange

Le temps d'échange est le délai écoulé entre la création d'un échange et la fin de ce dernier. Il inclut la durée de la communication sur la liaison série.

La création de l'échange se produit lors de l'appel de la fonction de communication.

La fin de l'échange se fait sur 1 des événements suivants :

- réception de données,
- Une anomalie se produit.
- temporisation écoulée.

Temps d'échange pour 1 mot

Le tableau ci-dessous montre les temps d'échange pour une communication Modbus à un mot sur un module BMX NOM 0200 :

Débit de la communication en bits/s	Temps de cycle en ms	Temps d'échange en ms L'esclave Modbus est un BMX P34 1000 cyclique
4800	Cyclique	65
4800	10	68
4800	50	100
9600	Cyclique	38
9600	10	47
9600	50	50
19200	Cyclique	29
19200	10	38
19200	50	50
38400	Cyclique	24
38400	10	30
38400	50	50
57600	Cyclique	17
57600	10	20

Débit de la communication en bits/s	Temps de cycle en ms	Temps d'échange en ms L'esclave Modbus est un BMX P34 1000 cyclique
57600	50	50
115200	Cyclique	17
115200	10	20
115200	50	50

Temps d'échange pour 100 mots

Le tableau ci-dessous montre les temps d'échange pour une communication Modbus à 100 mots sur un processeur BMX NOM 0200 :

Débit de la communication en bits/s	Temps de cycle en ms	Temps d'échange en ms L'esclave Modbus est un BMX P34 1000 cyclique
4800	Cyclique	560
4800	10	560
4800	50	600
9600	Cyclique	286
9600	10	295
9600	50	300
19200	Cyclique	152
19200	10	160
19200	50	200
38400	Cyclique	86
38400	10	90
38400	50	100
57600	Cyclique	56
57600	10	60
57600	50	100
115200	Cyclique	36
115200	10	40
115200	50	50

Précision de mesure

Tous les temps d'échange cités ci-dessus sont issus de mesures dont la précision est de +/-10 ms.

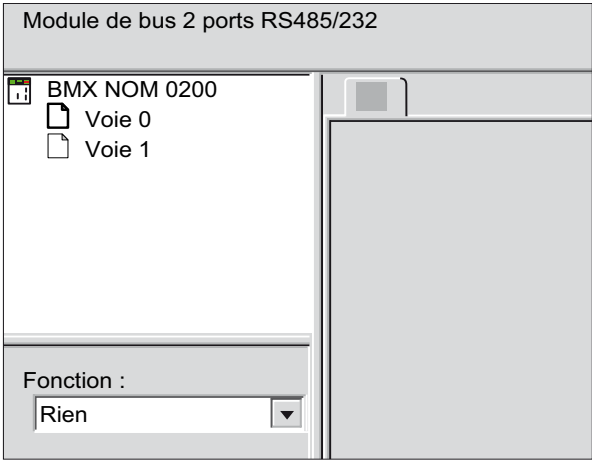
Comment accéder aux paramètres de la liaison série

Vue d'ensemble

Les pages suivantes expliquent comment accéder à l'écran de configuration du port série pour le module BMX NOM 0200. Vous y trouverez également des informations générales sur les écrans de configuration et de mise au point des liaisons Modbus et de mode caractère :

Comment accéder à la liaison série

Le tableau suivant présente la marche à suivre pour accéder à la liaison série d'un module BMX NOM 0200 :

Etape	Action
1	Ouvrez l'éditeur de configuration matérielle.
2	Double-cliquez sur le module BMX NOM 0200.
3	Sélectionnez la voie à configurer (voie 0 ou voie 1). Résultat avec la voie 0 sélectionnée : 

Etape	Action
4	Sélectionnez la fonction Liaison Modbus. Résultat avec la voie 0 sélectionnée : 2 Module de bus 2 ports RS485/232 3 BMX NOM 0200 Voe 0 Voe 1 Configuration Type Master Maitre Nombre de réitérations 3 Délai de réponse 100 X 10 ms Esclave Numéro d'esclave 1 Externe Ligne physique ☒ RS232 ☐ RS485 Signaux ☐ RX/TX ☐ RX/TX + RTS/CTS ☒ RX/TX + RTS/CTS + DTR/DSR/DCD Vitesse de transmission 19200 bits/s Délai entre trames ☒ Par défaut 2 ms Données ☐ ASCII (7 bits) ☒ RTU (8 bits) Arrêt ☐ 1 bit ☒ 2 bits Parité ☒ Paire ☐ Impaire ☐ Rien Retard RTC/CTS 0 X 100 ms

Description de l'écran de configuration

Le tableau suivant présente les différents éléments des écrans de configuration :

Touche	Élément	Fonction
1	Onglets	L'onglet en avant plan indique le mode utilisé (Configuration pour cet exemple). Chaque mode peut être sélectionné par l'onglet correspondant. Les modes disponibles sont : ● Configuration ● Mise au point (accessible uniquement en mode connecté), ● Diagnostic (accessible uniquement en mode connecté).
2	Zone du module	Affiche la référence du module et l'état des témoins du module en mode en ligne.
3	Zone de voie	Permet : ● d'afficher les onglets suivants en cliquant sur BMX NOM 0200 : ○ « Présentation » qui donne les caractéristiques de l'équipement, ○ « Objets d'E/S » (voir <i>Unity Pro, Modes de marche</i>) qui permet de présymboliser les objets d'entrées/sorties, ○ « Défaut », qui donne accès aux défaillances détectées de l'équipement (en mode connecté). ● d'afficher les onglets suivant en cliquant sur Voie 0 ou Voie 1 : ○ « Configuration » ○ « Mise au point » ○ « Défaut » ● d'afficher le symbole et le nom de la voie définis par l'utilisateur (à l'aide de l'éditeur de variables).
4	Zone des paramètres généraux	Permet de choisir les paramètres généraux associés à la voie : ● Fonction : les fonctions disponibles sont « Rien », « Liaison Modbus » et « Liaison mode caractère ». Par défaut, la fonction « Rien » est configurée. ● Tâche : définit la tâche maître dans laquelle seront échangés les objets à échange implicite de la voie. Cette zone est en grisé et ne peut pas être configurée.
5	Configuration, zone de mise au point ou défaut	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. En mode diagnostic, permet d'afficher les erreurs détectées en cours au niveau du module ou de la voie.

Sous-chapitre 7.2

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 des communications série Modbus dans un rack local Modicon M340	140
Ecran de configuration de la communication série Modbus du module BMX NOM 0200 dans une station X80	143
Fonctions Modbus accessibles	146
Valeurs par défaut des paramètres d'une communication Modbus Serial	147
Paramètres Modbus liés à l'application	148
Paramètres Modbus liés à la transmission	150
Paramètres liés aux signaux et à la ligne physique dans Modbus	152
Configuration de l'adresse de l'esclave MODBUS BMX NOM0200 sans Unity Pro	154

Ecran de configuration des communications série Modbus dans un rack local Modicon M340

Général

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

Accès à l'écran de configuration

Le tableau suivant présente la marche à suivre pour accéder à l'écran de configuration d'une communication Modbus Serial :

Etape	Action
1	Ouvrez le sous-répertoire BMX NOM 0200 dans le navigateur de projet (<i>voir page 136</i>).
2	Sélectionnez la voie à configurer et la fonction "Liaison Modbus" dans l'écran qui apparaît.

Illustration

La figure ci-dessous représente un écran de configuration par défaut d'une communication Modbus Serial sur la voie 0 :

The screenshot shows a configuration window titled 'Module de bus 2 ports RS2485/232'. On the left, a tree view shows 'BMX NOM 0200.2' with sub-items 'Voie 0' and 'Voie 1'. The 'Configuration' tab is active. The interface is divided into several sections:

- Type:** A dropdown menu set to 'Esclave'.
- Maître:** Includes 'Nombre de répétitions' (0) and 'Délai de réponse' (1 X 10 ms).
- Esclave:** Includes 'Numéro d'esclave' (1) and an 'Externe' checkbox.
- Vitesse de transmission:** A dropdown menu set to '19200 bits/s'.
- Délai entre les trames:** A checkbox 'Par défaut' is checked, with a value of 2 ms.
- Données:** Radio buttons for 'ASCII (7 bits)' and 'RTU (8 bits)' (selected).
- Stop:** Radio buttons for '1 bit' (selected) and '2 bits'.
- Parité:** Radio buttons for 'Paire' (selected), 'Impaire', and 'Aucune'.
- Retard RTS/CTS:** A value of 0 X 100 ms.
- Ligne physique:** Radio buttons for 'RS232' (selected) and 'RS485'.
- Signaux:** Radio buttons for 'RX/TX' (selected), 'RX/TX • RTS/CTS', and 'RX/TX • RTS/CTS • DTR/DSR/DCD'.

At the bottom left, there are fields for 'Fonction : Liaison Modbus' and 'Tâche : MAST'. The bottom status bar shows 'Bus automate' and '0.1 : BMX N...'.

Three blue numbers are overlaid on the image: '1' points to the 'Voie 0' item in the tree, '2' points to the 'Stop' section, and '3' points to the 'Ligne physique' section.

Description

Ces zones permettent de configurer les paramètres de voie. En mode connecté, ces zones sont accessibles. En mode local, elles sont accessibles, mais certains paramètres ne le sont pas. Ils sont alors grisés.

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

Touche	Élément	Commentaire
1	les paramètres concernant l'application, (voir page 148)	Ces paramètres sont accessibles à partir de 3 zones : ● Type, ● Maître, ● Esclave.
2	les paramètres concernant la transmission, (voir page 150)	Ces paramètres sont accessibles à partir de cinq zones : ● Vitesse de transmission, ● Délai entre les trames, ● Données, ● Stop bits, ● Parité.
3	les paramètres concernant les signaux et la ligne physique. (voir page 152)	Ces paramètres sont accessibles à partir de 3 zones : ● Ligne physique, ● Signaux, ● Retard RTS/CTS.

NOTE : Pour la configuration de la communication Modbus Serial en mode maître, la zone "esclave" est grisée et ne peut pas être modifiée, et vice-versa.

Ecran de configuration de la communication série Modbus du module BMX NOM 0200 dans une station X80

Général

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

REMARQUE : lorsque le module BMX NOM 0200 est une station X80 d'E/S Ethernet Quantum, il doit avoir les versions suivantes :

- Logiciel du produit = 04
- Logiciel du logiciel ≥ 1.4

Ces informations figurent sur l'étiquette située sur le côté du module.

Accès à l'écran de configuration

Le tableau suivant présente la marche à suivre pour accéder à l'écran de configuration d'une communication Modbus Serial :

Etape	Action
1	Ouvrez le sous-répertoire BMX NOM 0200 dans le navigateur de projet (<i>voir page 136</i>).
2	Sélectionnez la voie à configurer et la fonction "Liaison Modbus" dans l'écran qui apparaît.

Illustration

La figure ci-dessous représente un écran de configuration par défaut d'une communication Modbus Serial sur la voie 0 :

The screenshot shows a configuration window titled "Module de bus 2 ports RS2485/232". On the left, a tree view shows "BMX NOM 0200.3" with sub-items "Voie 0" and "Voie 1". The "Configuration" tab is active. The interface is divided into several sections:

- Type:** A dropdown menu set to "Esclave".
- Maître:** Includes "Nombre de répétitions" (0) and "Délai de réponse" (1 X 10 ms).
- Esclave:** Includes "Numéro d'esclave" (1) and an "Externe" checkbox.
- Vitesse de transmission:** A dropdown menu set to "19200 bits/s".
- Délai entre les trames:** A checkbox "Par défaut" is checked, with a value of 2 ms.
- Données:** Radio buttons for "ASCII (7 bits)" and "RTU (8 bits)".
- Stop:** Radio buttons for "1 bit" and "2 bits".
- Parité:** Radio buttons for "Paire", "Impaire", and "Aucune".
- Retard RTS/CTS:** A value of 0 X 100 ms.
- Ligne physique:** Radio buttons for "RS232" and "RS485".
- Signaux:** Radio buttons for "RX/TX", "RX/TX • RTS/CTS", and "RX/TX • RTS/CTS • DTR/DSR/DCD".

At the bottom left, there are fields for "Fonction:" (set to "Liaison Modbus") and "Tâche:" (set to "MAST"). The bottom status bar shows "Bus automate" and "0.1 : BMX N...".

Annotations: A blue line labeled "1" points to the "Voie 0" item in the tree view. A blue line labeled "2" points to the "Stop" section. A blue line labeled "3" points to the "Ligne physique" section.

Description

Ces zones permettent de configurer les paramètres de voie. En mode connecté, ces zones sont accessibles. En mode local, elles sont accessibles, mais certains paramètres ne le sont pas. Ils sont alors grisés.

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

Touche	Elément	Commentaire
1	les paramètres concernant l'application, (voir page 148)	Ces paramètres sont accessibles à partir de 3 zones : ● Type, ● Maître, ● Esclave.
2	les paramètres concernant la transmission, (voir page 150)	Ces paramètres sont accessibles à partir de cinq zones : ● Vitesse de transmission, ● Délai entre les trames, ● Données, ● Bits d'arrêt, ● Parité.
3	les paramètres concernant les signaux et la ligne physique. (voir page 152)	Ces paramètres sont accessibles à partir de 3 zones : ● Ligne physique, ● Signaux, ● Retard RTS/CTS.

NOTE : lorsque vous configurez la communication série Modbus en mode maître, les paramètres **Esclave** sont grisés et non modifiables.

Fonctions Modbus accessibles

Présentation

L'accessibilité des fonctions pour la configuration d'une liaison série d'un module BMX NOM 0200 utilisant Modbus Serial est fonction de la ligne physique utilisée.

Fonctions accessibles

Le tableau ci-dessous indique les différentes fonctions configurables selon le type de liaison série utilisée :

Fonction	Liaison RS485 (sur Voie 0 ou Voie 1)	Liaison RS232 (sur Voie 0)
Nombre de répétitions du maître	X	X
Délai de réponse du maître	X	X
Numéro d'esclave	X	X
Vitesse de transmission	X	X
Délai entre les trames	X	X
Données	● ASCII (7 bits) ● RTU (8 bits)	● ASCII (7 bits) ● RTU (8 bits)
Arrêter	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Parité	● Impaire ● Paire ● Rien	● Impaire ● Paire ● Rien
Signaux RX/TX	X	X
Signaux RTS/CTS	-	X
Retard RTS/CTS	-	X
Signaux DTR/DSR/DCD	-	X
Polarisation	-	-

- X Fonction accessible
- Fonction inaccessible

Valeurs par défaut des paramètres d'une communication Modbus Serial

Présentation

Les paramètres d'une communication Modbus Serial possèdent tous des valeurs par défaut.

Valeurs par défaut

Le tableau suivant présente les valeurs par défaut des paramètres d'une communication Modbus Serial sur Voie 0 et Voie 1 du module BMX NOM 0200 :

Paramètre de configuration	Valeur
Mode	Esclave
Tracé physique	RS232
Numéro d'esclave	1
Délai entre les trames	2 ms
Vitesse de transmission	19200 bits/s
Parité	Paire
Bits de données	RTU (8 bits)
Bits d'arrêt	1 bit

Paramètres Modbus liés à l'application

Présentation

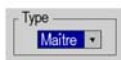
Après avoir configuré la voie de communication, vous devez renseigner les paramètres dédiés à l'application.

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 :

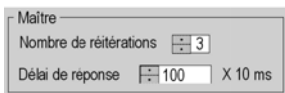


Cette zone permet de sélectionner le rôle à configurer pour le module dans la communication Modbus Serial :

- **Maître** : Dans le cas où le module est maître.
- **Esclave** : Dans le cas où le module 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.

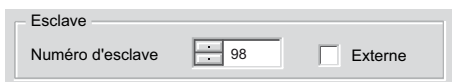
la valeur par défaut est de 1 seconde (100*10 ms),
les valeurs possibles sont comprises entre 10 ms et 10 s.

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.

NOTE : En mode diffusion, la valeur configurée dans le champ Délai de réponse est utilisée comme délai de diffusion : le temps minimum entre deux échanges en mode diffusion.

Zone Esclave

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



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.

La sélection de **Externe** grise le champ **Numéro d'esclave** et fait utiliser par le module la valeur de l'adresse d'esclave enregistrée dans sa (voir page 154) mémoire FLASH interne.

NOTE : si l'adresse enregistrée dans la mémoire FLASH n'est pas dans la plage d'adresse MODBUS, c'est l'adresse d'esclave par défaut 248 qui est utilisée.

Lors de la mise à jour du firmware du module, l'adresse d'esclave par défaut enregistrée dans la mémoire FLASH est réglée à 248. Une nouvelle commande doit être utilisée pour réinitialiser l'adresse dans la mémoire FLASH.

Paramètres Modbus liés à la transmission

Présentation

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

Ces paramètres sont accessibles à partir de 5 zones :

- Zone Vitesse de transmission,
- Zone Délai inter-caractères,
- Zone Données,
- Zone Stop,
- Zone Parité.

Zone Vitesse de transmission

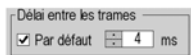
Cette zone de configuration apparaît dans l'écran comme représenté ci-dessous :



Cette zone permet de sélectionner la vitesse de transmission de la liaison Modbus Serial. Cette vitesse est en conformité avec les autres équipements. Les valeurs configurables sont 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 et 115200 bits/s (seulement sur la voie 0 en mode RS232).

Zone Délai entre les trames

Cette zone de configuration affichée ci-dessous n'est accessible qu'en mode RTU (elle est grisée en mode ASCII).



Le délai entre les trames est le délai minimal séparant 2 trames en réception. Ce délai est géré quand le BMX NOM 0200 est en réception de messages, qu'il soit maître ou esclave.

NOTE : la valeur par défaut dépend de la vitesse de transmission choisie.

NOTE : le délai entre les trames doit correspondre à la valeur par défaut pour être conforme au protocole Modbus. Si un esclave n'est pas conforme, la valeur peut être modifiée et doit être identique pour le maître et tous les esclaves du bus.

Zone Données

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



Cette zone permet de renseigner le type de codage utilisé pour communiquer avec la liaison Modbus Serial. Ce champ est à configurer en fonction des autres équipements connectés sur le bus. Deux modes sont configurables :

- Mode RTU :
 - les caractères sont codés sur 8 bits,
 - la fin de trame est détectée par un silence d'au moins 3,5 caractères,
 - l'intégrité de la trame est contrôlée à l'aide d'un mot appelé checksum CRC contenu dans celle-ci.
- Mode ASCII :
 - les caractères sont codés sur 7 bits,
 - le début de trame est détecté par réception du caractère ":",
 - la fin de trame est détectée par un retour chariot et un saut de ligne,
 - l'intégrité de la trame est contrôlée à l'aide d'un octet appelé checksum LRC contenu dans celle-ci.

Zone Stop

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



Cette zone permet de renseigner le nombre de bits d'arrêt utilisés pour communiquer. Ce champ est à configurer en fonction des autres équipements. Les valeurs configurables sont :

- 1 bit
- 2 bits

Zone Parité

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



Cette zone permet de définir l'adjonction ou non d'un bit de parité ainsi que son type. Ce champ est à configurer en fonction des autres équipements. Les valeurs configurables sont :

- Paire
- Impaire
- Rien

Paramètres liés aux signaux et à la ligne physique dans Modbus

Vue d'ensemble

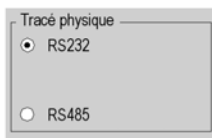
Après avoir configuré la voie de communication, vous devez renseigner les paramètres de la ligne physique et des signaux.

Ces paramètres sont accessibles à partir de trois zones :

- zone Ligne Physique,
- zone Signaux,
- zone Retard RTS/CTS.

Zone Ligne Physique

Cette zone de configuration affichée ci-dessous n'est accessible que sur la voie 0 (elle est grisée et configurée à RS485 sur la voie 1) :

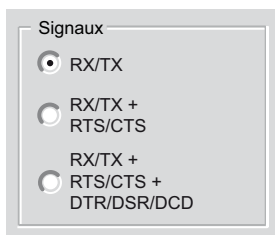


Cette zone permet de choisir le type de la ligne physique du port série du module BMX NOM 0200 parmi les deux types suivants :

- Ligne RS232,
- Ligne RS485.

Zone Signaux

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



Cette zone permet de sélectionner les signaux supportés par la ligne physique RS232 :

- RX/TX
- RX/TX + RTS/CTS (signaux de gestion de contrôle de flux matériel)
- RX/TX + RTS/CTS + DTR/DSR/DCD (signaux de modem)

Si la ligne configurée est de type RS 485, la zone est entièrement grisée et la valeur par défaut est RX/TX.

Zone retard RTS/CTS

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



La zone de temporisation RTS/CTS n'est disponible que quand les deux cases RS232 et RX/TX+RTS/CTS ou RX/TX+RTS/CTS+DTR/DSR/DCD sont cochées. Un contrôle de flux matériel RTS/CTS est utilisé.

L'algorithme de contrôle de flux matériel RTS/CTS vise à éviter le débordement de tampon de réception (full duplex).

La temporisation RTS/CTS correspond au délai d'attente entre la montée de RTS et la montée de CTS. Une valeur de temporisation RTS/CTS différente de 0 correspond aussi au temps d'attente maximum entre chaque transmission de caractère après la montée des signaux RTS et CTS. Si la valeur est réglée sur 0, les puces UART peuvent rester coincées en état d'attente pendant un temps infini jusqu'à la montée de CTS, donc la valeur 0 n'est utilisée que dans des cas particuliers tels que bouclage du signal RTS sur le signal CTS de façon à vérifier qu'aucun câble de connexion n'est défectueux.

NOTE : la valeur par défaut est 0 ms.

Configuration de l'adresse de l'esclave MODBUS BMX NOM0200 sans Unity Pro

Conditions requises

L'adresse FLASH peut être mise à jour dans tous les modes mais n'est prise en compte que lorsqu'un mode de fonctionnement est activé.

La liste ci-dessous indique les conditions requises pour configurer l'adresse du MODBUS BMX NOM0200 sans Unity Pro :

- Pour utiliser l'adresse FLASH, le module doit être configuré :
 - dans le protocole de l'esclave MODBUS avec la case à cocher **EXTERNE** ;
 - dans le protocole du maître MODBUS ou en mode **CHAR** suivi d'un basculement vers le protocole de l'esclave MODBUS.

Mise à jour de l'adresse de l'esclave MODBUS dans la mémoire FLASH par des commandes applicatives

Le tableau ci-dessous explique comment mettre à jour l'adresse de l'esclave MODBUS dans la mémoire FLASH à l'aide de commandes applicatives :

Etape	Action
1	Stockez l'adresse de l'esclave dans %MWr . m . c . 25.
2	Mise à 1 du bit %MWr . m . c . 24 . 7.
3	Envoyez la commande WRITE_CMD à la voie du module.
4	Vérifiez que la commande est exécutée (mise à 0 de %MWr . m . c . 0 . 1) et acceptée (une valeur nulle de %MWr . m . c . 1 . 1 signifie aucune erreur) => la mémoire FLASH est mise à jour.
5	Activez l'un des modes de fonctionnement suivants sur la voie pour prendre en compte la nouvelle adresse : ● Chargement des applications ● Démarrage à froid ● Démarrage à chaud ● Remplacement à chaud ● Changement de protocole (VERS ESCLAVE)
6	Exécutez la commande READ_STS sur la voie pour vérifier l'adresse de l'esclave dans l'octet de poids fort de %MWr . m . c . 3.

NOTE : une même commande peut inclure plusieurs ordres. Si l'un d'entre eux ne peut pas être exécuté, c'est la commande en totalité qui est rejetée et aucun ordre n'est exécuté.

Mise à jour de l'adresse de l'esclave MODBUS dans la mémoire FLASH via la ligne série

Le tableau ci-dessous explique comment mettre à jour l'adresse de l'esclave MODBUS dans la mémoire FLASH via la ligne série :

Etape	Action
1	Configurez l'équipement MAÎTRE avec le même paramètre de ligne série qu'une voie du module.
2	Connectez le MAÎTRE au module en mode point-à-point.
3	Envoyez la requête 0x11 à l'adresse point-à-point : 0xF8 0x11 0x01 numéro_voie(0 ou 1) slaveID(0 . . 0xF8)
4	Vérifiez que la réponse est OK => la mémoire FLASH est mise à jour.
5	Activez un mode de fonctionnement sur la voie pour prendre en compte la modification apportée à l'étape 4.
6	Envoyez une requête 0x11 pour vérifier la nouvelle adresse de l'esclave : slaveID 0x11 0x01

NOTE : évitez de modifier la mémoire FLASH régulièrement pour ne pas endommager ce composant (100 000 cycles d'écriture au maximum).

Sous-chapitre 7.3

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 module maître d'une liaison Modbus	157
Services pris en charge par le module esclave d'une liaison Modbus	165
Informations sur le mode expert Modbus	167

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

Présentation

S'il est utilisé en tant que maître sur une liaison Modbus, un module BMX NOM 0200 assure la prise en charge de plusieurs services par les fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH.

Echanges de données

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

Ces requêtes utilisent les fonctions de communication READ_VAR et WRITE_VAR :

Requête Modbus	Code fonction	Fonction de communication
Lecture de bits	16#01 ou 16#02	READ_VAR
Lecture de mots	16#03 ou 16#04	READ_VAR
Ecriture de bits	16#0F	WRITE_VAR
Ecriture de mots	16#10	WRITE_VAR

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

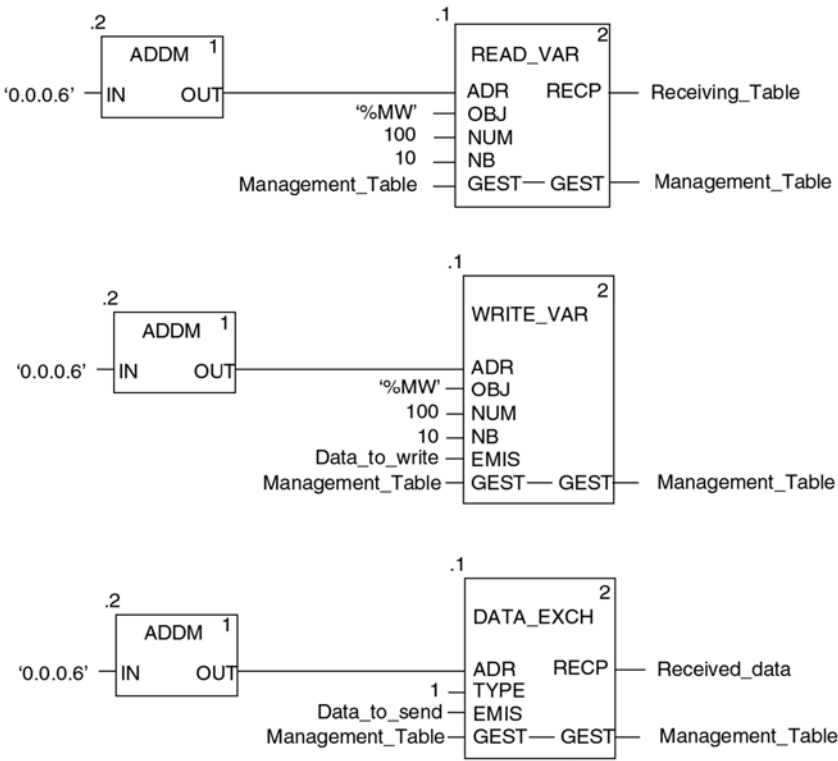
Fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH

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

- READ_VAR : lecture de variables.
- WRITE_VAR : écriture de variables.
- DATA_EXCH : Pour envoyer des requêtes Modbus à un autre appareil par le protocole sélectionné

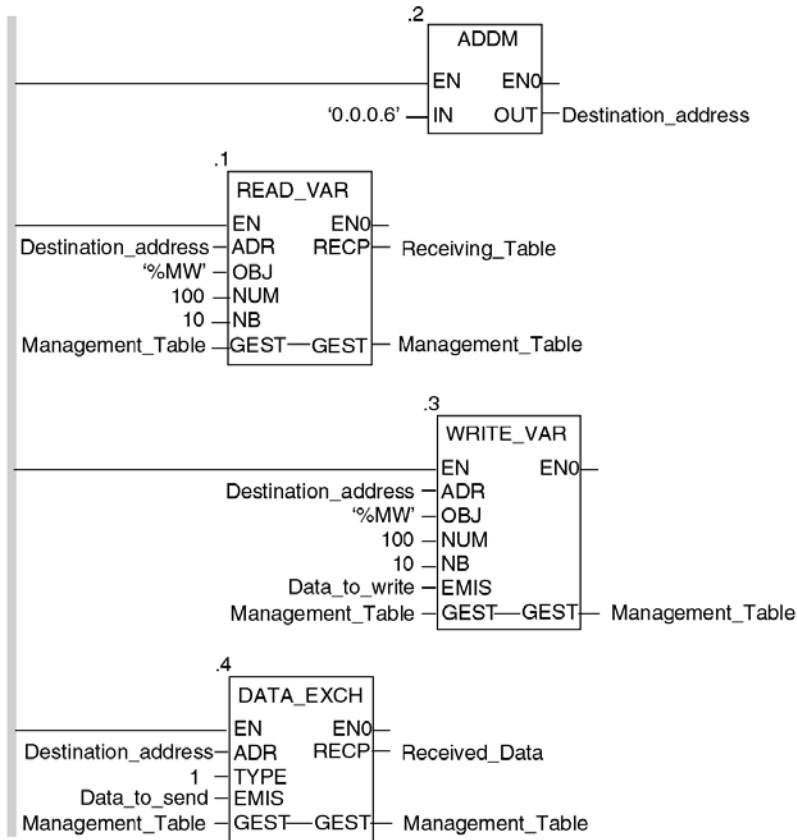
Exemple de programmation en FBD

Le schéma ci-dessous représente un exemple de programmation des fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH dans le langage FBD :



Exemple de programmation en Ladder

La figure suivante représente un exemple de programmation en langage Ladder des fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH :



Exemple de programmation en ST

Les lignes de code ci-dessous représentent un exemple de programmation des fonctions de communication READ_VAR, WRITE_VAR et DATA_EXCH dans le langage ST :

```
READ_VAR(ADDM('0.0.0.6'), 'MW', 100, 10, Management_Table,  
Receiving_Table);  
  
WRITE_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, Data_to_write,  
Management_Table);  
  
DATA_EXCH(ADDM('0.0.0.6'), 1, Data_to_send, Management_Table,  
Received_data);
```

Annulation d'un échange

Un échange exécuté par les fonctions READ_VAR, WRITE_VAR et DATA_EXCH peut être annulé par l'une ou l'autre méthode de programmation, toutes les deux présentées en langage ST ci-dessous :

- utilisation de la fonction CANCEL :

```
IF (%MW40.0) THEN
  %MW200 := SHR(%MW40, 8);
  CANCEL (%MW200, %MW185);
END_IF;
```

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction READ_VAR et est à 1 lorsque la fonction de communication est active. Si ce bit est à 1, le programme exécute les instructions suivantes :

- décale à droite les bits de %MW40 d'un octet (8 bits) et charge l'octet correspondant au numéro d'échange de la communication dans le mot %MW200,
- annule l'échange dont le numéro d'échange est contenu dans le mot %MW200 via la fonction CANCEL.

- utilisation du bit d'annulation de la fonction de communication :

```
IF (%MW40.0) THEN
  SET(%MW40.1);
  READ_VAR(ADDM('0.0.0.6'), '%MW', 100, 10, %MW40:4, %MW10:10);
END_IF;
```

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction READ_VAR et est à 1 lorsque la fonction de communication est active. Si ce bit est défini sur 1, le programme met à 1 le bit %MW40.1, qui correspond au bit d'annulation de la fonction. Ceci a pour effet d'arrêter la communication de la fonction READ_VAR.

NOTE : Lors de l'utilisation du bit d'annulation de la fonction de communication contenu dans le mot de gestion d'échanges de la fonction (%MW40 dans cet exemple), la fonction (READ_VAR dans cet exemple) doit être appelée dans le but d'activer l'annulation de l'échange.

NOTE : Lors de l'utilisation du bit d'annulation de la fonction de communication, il est possible d'annuler une communication à partir d'une table d'animation. Pour cela, il suffit de mettre à 1 le bit d'annulation de la fonction (%MW40.1 dans cet exemple), puis de redémarrer la fonction de communication.

NOTE : Cet exemple de programmation concerne la fonction READ_VAR, mais il est tout aussi applicable aux fonctions WRITE_VAR ainsi que DATA_EXCH.

NOTE : l'utilisation de la fonction CANCEL utilise un mot de compte rendu pour la fonction CANCEL (%MW185 pour cet exemple).

Description des paramètres de la fonction ADDM

Le tableau suivant décrit les différents paramètres de la fonction ADDM :

Paramètre	Type	Description
IN	STRING	Adresse d'un équipement sur un bus ou sur une liaison série. La syntaxe de l'adresse est du type "r.m.c.node". L'adresse est composée des paramètres suivants : ● r : numéro de rack du module ● m : emplacement du module dans le rack ● c : numéro de voie du module ● node : numéro de l'esclave destinataire de la requête.
OUT	ARRAY [0..7] OF INT	Tableau représentant l'adresse d'un équipement. Ce paramètre est utilisable par plusieurs fonctions de communication comme paramètre d'entrée.

Description des paramètres de la fonction READ_VAR

Le tableau suivant décrit les différents paramètres de la fonction READ_VAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse de l'entité destinataire donnée par le paramètre OUT de la fonction ADDM.
OBJ	STRING	Type d'objet à lire. Les types disponibles sont les suivants : ● %M : bit interne, ● %MW : mot interne, ● %I : bit d'entrée externe, ● %IW : mot d'entrée externe.
NUM	DINT	Adresse du premier objet à lire.
NB	INT	Nombre d'objets consécutifs à lire.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● mot de rang 1 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Numéro d'échange, ○ Octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1). ● mot de rang 2 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Compte rendu d'opération, ○ Octet de poids faible : Compte rendu de communication. ● mot de rang 3 : mot géré par l'utilisateur et qui définit le temps d'attente maximal de réponse sur une base de 100 ms. ● mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.
RECP	ARRAY [n..m] OF INT	Tableau de mots contenant la valeur des objets lus.

Description des paramètres de la fonction WRITE_VAR

Le tableau suivant décrit les différents paramètres de la fonction WRITE_VAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse de l'entité destinataire donnée par le paramètre OUT de la fonction ADDM.
OBJ	STRING	Type d'objet à écrire. Les types disponibles sont les suivants : ● %M : bit interne, ● %MW : mot interne, Remarque : WRITE_VAR ne peut pas être utilisée pour les variables %I et %IW.
NUM	DINT	Adresse du premier objet à écrire.
NB	INT	Nombre d'objets consécutifs à écrire.
EMIS	ARRAY [n..m] OF INT	Tableau de mots contenant la valeur des objets à écrire.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● mot de rang 1 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Numéro d'échange, ○ Octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1). ● mot de rang 2 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Compte rendu d'opération, ○ Octet de poids faible : Compte rendu de communication. ● mot de rang 3 : mot géré par l'utilisateur et qui définit le temps d'attente maximal de réponse sur une base de 100 ms. ● mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.

Description des paramètres de la fonction DATA_EXCH

Le tableau suivant décrit les différents paramètres de la fonction DATA_EXCH :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse de l'entité destinataire donnée par le paramètre OUT de la fonction ADDM.
TYPE	INT	Pour les automates Modicon M340, la seule valeur possible est 1 : Envoi d'un tableau EMIS, l'automate attend ensuite la réception d'un tableau RECP.
EMIS	ARRAY [n..m] OF INT	Tableau d'entiers à envoyer à l'appareil destination de la requête. Remarque : Il est impératif que la longueur des données à envoyer (en octets) soit affectée au quatrième mot du tableau de gestion avant le lancement de la fonction, pour que celle-ci puisse être exécutée correctement.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● mot de rang 1 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Numéro d'échange. ○ Octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1). ● mot de rang 2 : mot géré par le système et composé de 2 octets : ○ Octet de poids fort : Compte rendu d'opération, ○ Octet de poids faible : Compte rendu de communication. ● mot de rang 3 : mot géré par l'utilisateur et qui définit le temps d'attente maximal de réponse sur une base de 100 ms. ● mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.
RECP	ARRAY [n..m] OF INT	Tableau d'entiers contenant les données reçues. Remarque : La taille des données reçues (en octets) est automatiquement écrite par le système dans le quatrième mot du tableau de gestion.

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

Présentation

En cas d'utilisation comme esclave dans une liaison Modbus, un module BMX NOM 0200 peut prendre en charge plusieurs services.

Echanges de données

Un module esclave gère les requêtes suivantes :

Requête Modbus	Code fonction	Objet automate
Lecture de n bits de sortie	16#01	%M
Lecture de n mots de sortie	16#03	%MW
Ecriture de n bits de sortie	16#0F	%M
Ecriture de n mots de sortie	16#10	%MW
Lecture/écriture de n mots de sortie	16#17	%MW

NOTE : Lecture/écriture de plusieurs %MW

L'opération WRITE s'exécute avant l'opération READ pour pouvoir lire et modifier les mêmes registres en même temps que I/O scanning. Si la taille d'échange du WRITE ou du READ est hors limites, l'état renvoyé est « ADRESSE DE DONNEES INCORRECTE », mais si seul le READ échoue, l'opération WRITE s'exécute avec le même état.

Diagnostic et maintenance

Les requêtes de diagnostics et maintenance gérées par un module BMX NOM 0200 esclave Modbus sont mentionnées ci-dessous :

Désignation	Code fonction / Code sous-fonction
Read exception status	16#07
Restart Communications Option	16#08 / 16#01
Return Diagnostic Register	16#08 / 16#02
Change ASCII Input Delimiter	16#08 / 16#03
Force Listen Only Mode	16#08 / 16#04
Clear Counters and Diagnostic Register	16#08 / 16#0A
Return Bus Message Count	16#08 / 16#0B
Return Bus Communication Error Count	16#08 / 16#0C
Return Bus Communication Error Count	16#08 / 16#0D
Return Slave Message Count	16#08 / 16#0E
Return Slave No Response Count	16#08 / 16#0F

Désignation	Code fonction / Code sous-fonction
Return Slave Negative Acknowledgements Count	16#08 / 16#10
Return Slave Busy Count	16#08 / 16#11
Return Bus Character Overrun Count	16#08 / 16#12
Get Communication event Counter	16#0B
Get Communication event Log	16#0C
Report Slave identification	16#11
Write Slave identification	16#11 / 16#01

Informations sur le mode expert Modbus

Communication en mode expert

Le mode expert est un ensemble de commandes qui peut être envoyé au module pour obtenir des fonctionnalités supplémentaires.

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr.m.c.24	CONTROL	Explicite	INT	Signal de commande, changer de protocole
%MWr.m.c.24.0		Explicite	BOOL	Effacer les compteurs locaux
%MWr.m.c.24.1		Explicite	BOOL	Modifier dynamiquement le compte de tentatives en mode maître MODBUS (%MW26)
%MWr.m.c.24.2		Explicite	BOOL	Modifier le délai de réponse des esclaves (%MW28) pour un esclave en particulier (%MW27) en mode maître
%MWr.m.c.24.3		Explicite	BOOL	Modifier le délai par défaut de non prise en compte des esclaves, le caractère reçu indiquant que les esclaves sont ignorés après une réception de trame transmise à l'UC (%MW29)
%MWr.m.c.24.4		Explicite	BOOL	Modifier les synchronisations internes MODBUS RTU t1,5ch (%MW31), t3,5ch (%MW30) et le délai entre commutateurs (%MW32). La mise à jour de cette valeur risque de perturber le module si celui-ci est en cours de fonctionnement.

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr.m.c.24.6		Explicite	BOOL	Modifier le mode de gestion du modem : HALF/FULL DUPLEX ● Si le bit est défini en même temps que RTS_ON (%MWr.m.c.24.10) fonctionne aussi avec RTS_OFF %MWr.m.c.24.11 et utilise DTR si %MWr.m.r.24.8 ou %MWr.m.r.24.9 est utilisé) le mode de modem half duplex est activé. ● Si ce bit est défini mais qu'aucun bit RTS ou DTR (%MWr.m.c.24.8, %MWr.m.c.24.9, %MWr.m.c.24.10, %MWr.m.c.24.11) ne l'est, le mode full duplex est activé. Le bit %MW26 sert à définir le délai de début (StartDelay) et le bit %MW27 le délai de fin (EndDelay). Les bits %MW24.5, %MW24.1 et %MW24.2 ne peuvent donc pas être utilisés simultanément. REMARQUE : l'utilisateur peut être amené à rétablir l'état correct des signaux RTS/DTR après que la commande a été acceptée.
%MWr.m.c.24.7	SAVE_SLAVE_ADDR	Explicite	BOOL	Enregistrer l'adresse esclave Modbus dans la mémoire FLASH (%MW25).
%MWr.m.c.24.8	DTR_ON	Explicite	BOOL	Définir le signal DTR (tension positive)
%MWr.m.c.24.9	DTR_OFF	Explicite	BOOL	Réinitialiser le signal DTR (tension négative)
%MWr.m.c.24.10		Explicite	BOOL	Définir le signal RTS (tension positive)

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr.m.c.24.11		Explicite	BOOL	Réinitialiser le signal RTS (tension négative)
%MWr.m.c.24.12	TO_MODBUS_MASTER	Explicite	BOOL	Basculer en mode maître
%MWr.m.c.24.13	TO_MODBUS_SLAVE	Explicite	BOOL	Basculer en mode esclave
%MWr.m.c.24.14	TO_CHAR_MODE	Explicite	BOOL	Basculer en mode caractère
%MWr.m.c.25	SLAVE_ADDR	Explicite	INT	Adresse esclave Modbus pour le stockage dans la mémoire FLASH
%MWr.m.c.26		Explicite	INT	LOW BYTE : compte de tentatives en mode maître (MasterRetries) : nombre de tentatives en mode maître [0 à 15], voir %MW24.1 Délai de début (StartDelay) si le bit %MW26.6 est défini. Délai à respecter lorsque le CTS est OK avant de commencer à envoyer la trame. Ce bit est utile pour les modems qui demandent un délai supplémentaire après le CTS ou qui ne gèrent pas le signal CTS (dans ce cas, le RTS doit être connecté au CTS). Ce délai est exprimé en millisecondes, avec une précision de 3 ms environ. Il ne peut être défini qu'en mode RS232.

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr . m . c . 27		Explicite	INT	LOW BYTE : esclave pour lequel le maître adaptera le délai de réponse [0 à 248, 255=ALL], voir %MW24 . 2 et %MW28 Délai de fin (EndDelay) si le bit %MW24 . 6 est défini. Délai à respecter après l'envoi d'une trame, avant d'émettre le signal RTS de façon à donner suffisamment de temps au MODEM pour qu'il envoie la trame avant la fin de communication. Ce délai est exprimé en millisecondes, avec une précision de 3 ms environ. Il ne peut être défini qu'en mode RS232.
%MWr . m . c . 28		Explicite	INT	Délai de réponse associé à un esclave en particulier par tranche de 10 ms [1 à 1000], voir %MW24 . 2 et %MW27
%MWr . m . c . 29		Explicite	INT	Délai de non-prise en compte par tranche de 10 ms [1 à 10], voir %MW24 . 3
%MWr . m . c . 30		Explicite	INT	T3,5char : délai intertrames en millisecondes [0 à 10000]. La valeur utilisée dépend de la vitesse. Si elle est inférieure ou supérieure aux valeurs possibles, la limite inférieure ou supérieure est appliquée, et la commande est acceptée. La valeur 0 signifie que la RTU n'a subi aucune transformation. Le délai de réponse est recalculé.

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr . m . c . 31		Explicite	INT	T1,5char : délai entre caractères en millisecondes [0 à 9999]. La valeur utilisée dépend de la vitesse. Si elle est inférieure ou supérieure aux valeurs possibles, la limite inférieure ou supérieure est appliquée, et la commande est acceptée. La valeur 0 signifie que T1,5 est calculé comme T3,5ch – 2ch (calcul par défaut).
%MWr . m . c . 32		Explicite	INT	Délai maître entre commutateurs en mode RTU [0 à 256] en millisecondes. La valeur 0 signifie « aucun délai » ; si la valeur est inférieure à 10 bits, la valeur minimale de 10 bits est utilisée.

Exemple de code

```
(* le module NOM côté maître se trouve dans le rack 0 à l'emplacement 9 *)
if HalfModemMaster then
  HalfModemMaster:=false;
  %MW0.9.0.24:=16#0450;(* basculer en mode half duplex avec RTS, et modifier les synchroni-
sations MODBUS *)
  %MW0.9.0.26:=12;(* 12 ms à attendre avant de procéder à l'envoi quand CTS est activé *)
  %MW0.9.0.27:=9; (* laisser RTS activé pendant 9 ms après la fin de l'envoi *)
  %MW0.9.0.30:=0;
  %MW0.9.0.31:=0;(* utiliser la valeur de l'écran de configuration, soit 6 ms *)
  %MW0.9.0.32:=50; (* délai de 50 ms à respecter avant l'envoi d'une nouvelle trame *)
  write_cmd(%ch0.9.0);(* envoyer une commande et des données à la voie NOM *)
end_if;
(* côté esclave, le NOM se trouve dans le rack 0, à l'emplacement 3 *)
if HalfModemSlave then
  HalfModemSlave:=false;
```

```

%MW0.3.0.24:=16#0448;(* basculer en mode half duplex avec RTS, et modifier le temps de non
prise en compte des esclaves *)
%MW0.3.0.26:=12;(* attendre 12 ms avant de procéder à l'envoi quand CTS est activé *)
%MW0.3.0.27:=9; (* laisser le RTS activé pendant 9 ms après la fin de l'envoi *)
%MW0.3.0.29:=4; (* délai de non prise en compte à respecter : 4*10 ms *)
write_cmd(%ch0.3.0);(* envoyer une commande et des données à la voie NOM *)
end_if;
(* facultatif : envoi automatique de la commande *)
if %S0 or %S1 or %S13 then
    memoSendCmd:=true;
end_if;
(* copier chaque cycle d'erreur de module pour détecter une éventuelle disparition de celui-ci *)
memoSendCmd:=%I0.3.0.ERR;
(* si le module est OK, envoyer la commande une fois *)
if FE(memoSendCmd) then
    HalfModemSlave:=true;
end_if;

```

Accès aux registres internes NOM

Pour accéder aux registres internes Nom, il faut activer le mode MODBUS et utiliser l'EF READ_VAR. Exemple de code (le modèle NOM se trouve dans le rack 0, à l'emplacement 3) :

```

if dataCh030GetChannelGlobalInfo then
    read_var(addm('0.3.0'), '%MW', 200, 3, dataCh030Mgt, dataCh030Buff);
(* Internal_Reg@200 sont copiés dans le tampon dataCh030Buff *)
    dataCh030GetChannelGlobalInfo := false;
end_if;

```

- Internal_Reg@0 : délai de début (StartDelay) en ms (précision de 3 ms environ) (accès en lecture ou en écriture)
- Internal_Reg@1 : délai de fin (EndDelay) en ms (précision de 3 ms environ) (accès en lecture ou en écriture)
- Internal_Reg@200 : numéro de version d'interface = 1
- Internal_Reg@201 : adresse esclave stockée dans la mémoire FLASH
- Internal_Reg@202 : 1 = modification de la mémoire FLASH possible, 0= modification impossible
- Internal_Reg@1000 : code interne RTU maître ch0=1110, ch1=2110

- Internal_Reg@1002 : 0 = Full Duplex - commande de flux matériel, ou RS485 ; 1 = Half Duplex - direction gérée automatiquement par le module avec RTS
- Internal_Reg@1010 : délai interne en bits de temporisation de sortie des caractères (nbre de bits*1000/vitesse => durée en ms) [T1,5S]
- Internal_Reg@1012 : délai interne en bits de temporisation d'entrée des caractères [T1,5R]
- Internal_Reg@1014 : délai interne en bits de temporisation de sortie des trames [T3,5S]
- Internal_Reg@1016 : délai interne en bits de temporisation d'entrée des trames [T3,5R]
- Internal_Reg@1018 : délai en bits à respecter avant l'envoi des prochaines trames
- Internal_Reg@1090 : compte de tentatives maîtres (MasterRetries)
- Internal_Reg@1100 : délai de réponse esclave par tranche de 10 ms pour la diffusion
- Internal_Reg@1101 : délai de réponse esclave par tranche de 10 ms pour l'esclave 1
- ...
- Internal_Reg@1348 : délai de réponse esclave pour l'adresse point à point (248)
- Internal_Reg@1500 : code interne Modbus RTU esclave ch0=1120, ch1=2120
- Internal_Reg@1502 : 0 = Full Duplex - commande de flux matériel, ou RS485 ; 1 = Half Duplex - direction gérée automatiquement par le module avec RTS
- Internal_Reg@1510 : délai en bits de temporisation de sortie des caractères (nbre bits*1000/vitesse => durée en ms) [T1,5S]
- Internal_Reg@1512 : délai interne en bits de temporisation d'entrée des caractères [T1,5R]
- Internal_Reg@1514 : délai interne en bits de temporisation de sortie des trames [T3,5S]
- Internal_Reg@1516 : délai interne en bits de temporisation d'entrée des trames [T3,5R]
- Internal_Reg@1518 : délai en bits à respecter avant l'envoi des prochaines trames
- Internal_Reg@1602 : délai en ms de non-prise en compte après réception
- Internal_Reg@1606 : mode Ecoute seulement actif = 1, (non actif = 0)
- Internal_Reg@2000 : code interne Modbus ASCII maître ch0=1210, ch1=2210
- Internal_Reg@2002 : 0 = Full Duplex - commande de flux matériel, ou RS485 ; 1 = Half Duplex - direction gérée automatiquement par le module avec RTS
- Internal_Reg@2010 : délai en bits de temporisation de sortie des caractères (nbre bits*1000/vitesse => durée en ms) [T1,5S]
- Internal_Reg@2012 : délai interne en bits de temporisation d'entrée des caractères [T1,5R]
- Internal_Reg@2014 : délai interne en bits de temporisation de sortie des trames [T3,5S]
- Internal_Reg@2014 : délai interne en bits de temporisation de sortie des trames [T3,5S]
- Internal_Reg@2014 : délai interne en bits de temporisation de sortie des trames [T3,5S]
- Internal_Reg@2016 : délai interne en bits de temporisation d'entrée des trames [T3,5R]
- Internal_Reg@2018 : délai en bits à respecter avant l'envoi des prochaines trames
- Internal_Reg@2090 : compte de tentatives maîtres (MasterRetries)
- Internal_Reg@2100 : délai de réponse esclave par tranche de 10 ms pour la diffusion
- Internal_Reg@2101 : délai de réponse esclave par tranche de 10 ms pour l'esclave 1
- ...
- Internal_Reg@2348 : délai de réponse esclave pour l'adresse point à point (248)
- Internal_Reg@2500 : code interne Modbus ASCII esclave ch0=1220, ch1=2220
- Internal_Reg@2502 : 0 = Full Duplex - commande de flux matériel, ou RS485 ; 1 = Half Duplex - direction gérée automatiquement par le module avec RTS
- Internal_Reg@2510 : délai en bits de temporisation de sortie des caractères (nbre bits*1000/vitesse => durée en ms) [T1,5S]

-
- Internal_Reg@2512 : délai interne en bits de temporisation d'entrée des caractères [T1,5R]
 - Internal_Reg@2514 : délai interne en bits de temporisation de sortie des trames [T3,5S]
 - Internal_Reg@2516 : délai interne en bits de temporisation d'entrée des trames [T3,5R]
 - Internal_Reg@2518 : délai en bits à respecter avant l'envoi des prochaines trames
 - Internal_Reg@2600 : Adresse esclave en cours d'utilisation
 - Internal_Reg@2602 : délai en ms de non-prise en compte après réception
 - Internal_Reg@2606 : mode Ecoute seulement actif = 1, (non actif = 0)
 - Internal_Reg@3000 : code interne en mode caractère ch0=1000, ch1=2000
 - Internal_Reg@3002 : 0 = Full Duplex - commande de flux matériel, ou RS485 ; 1 = Half Duplex - direction gérée automatiquement par le module avec RTS
 - Internal_Reg@3100 : 0 = aucun critère d'arrêt actif, 1 = arrêt en cas de silence ou arrêt au niveau du caractère de fin
 - Internal_Reg@3102 : silence interne en bits (le minimum est 2 bits, le maximum 65535 bits)
 - Internal_Reg@3104 : premier octet de fin de trame à utiliser , 16#0100 signifie « aucun octet »
 - Internal_Reg@3106 : premier octet de fin de trame (EOF) : 1 = octet de fin de trame à laisser dans la trame, 0 = supprimer l'octet de fin de trame
 - Internal_Reg@3108 : deuxième octet de fin de trame
 - Internal_Reg@3110 : deuxième EOF : 1 = octet de fin de trame à laisser dans la trame, 0 = supprimer l'octet de fin de trame

Sous-chapitre 7.4

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. (voir page 140)
2	Sélectionnez l'onglet « Mise au point » dans l'écran qui apparaît.

Description de l'écran de mise au point

L'écran de mise au point se décompose en deux ou trois zones :

- Type et numéro de zone de l'esclave,
- la zone Compteurs,
- la zone Signaux (si RS232).

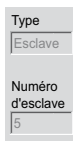
Type et numéro de zone de l'esclave

Si le module a la fonction de maître sur la liaison Modbus, cette zone se présente comme suit :



A screenshot of a graphical user interface element. It consists of a small rectangular box with a light gray background. At the top, the word 'Type' is written in a small, dark font. Below it, the word 'Maître' is displayed in a slightly larger, dark font, indicating it is the selected option.

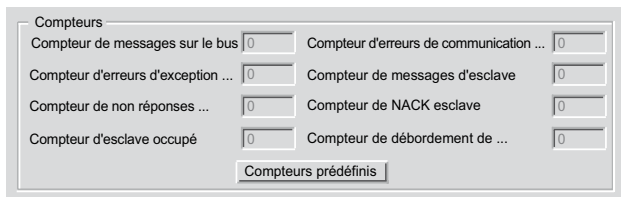
Si le module a la fonction d'esclave sur la liaison Modbus, cette zone se présente comme suit :



A screenshot of a graphical user interface element. It consists of two stacked rectangular boxes with light gray backgrounds. The top box has the word 'Type' at the top and 'Esclave' below it. The bottom box has the text 'Numéro d'esclave' at the top and the number '5' below it.

Zone Compteurs

La zone se présente comme ci-dessous :



Cette zone affiche les différents compteurs de mise au point.

Le bouton R.A.Z. 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 sont :

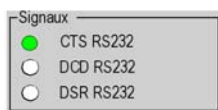
- **Compteur de messages sur le bus** : nombre de messages que le module a détectés sur la liaison série. Les messages dont la vérification CRC est négative ne sont pas pris en compte.
- **Compteur d'erreurs de communication sur le bus** : nombre de vérifications CRC négatives comptées par le module. 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** : nombre d'erreurs d'exception Modbus détectées par le module.
- **Compteur de messages d'esclaves** : nombre de message reçus et traités par la liaison Modbus,
- **Compteur de non réponses d'esclave(s)** : nombre de messages émis par le système distant pour lesquels il ne retourne pas de réponse (ni une réponse normale ni une réponse d'exception). Ce compteur compte aussi le nombre de messages reçus en mode diffusion.
- **Compteur d'accusés de réception négatifs d'esclaves** : nombre de messages adressés au système distant pour lesquels il retourne un accusé de réception négatif,
- **Compteur d'esclaves occupés** : nombre de messages adressés au système distant pour lesquels il retourne une réponse d'exception esclave occupé,
- **Compteur de dépassements de caractères sur le bus** : nombre de messages adressés au module qu'il ne peut pas acquérir à cause d'un dépassement de caractères. Un 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 un dysfonctionnement de matériel.

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 module.

Zone Signaux

Cette zone n'apparaît que si RS232 est sélectionné dans l'écran de configuration. Si RS485 est sélectionné dans l'écran de configuration, cette fenêtre n'apparaît pas du tout.

La zone Signaux se présente comme suit :



Cette zone indique l'activité des signaux :

- **CTS RS232** : indique l'activité du signal CTS.
- **DCD RS232** : indique l'activité du signal DCD.
- **DSR RS232** : indique l'activité du signal DSR.

Chapitre 8

Communication en Mode caractère du BMX NOM 0200

Objet de cette section

Ce chapitre vous présente la mise en œuvre logicielle de la communication en Mode caractère du BMX NOM 0200.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
8.1	Généralités	180
8.2	Configuration d'une communication en mode caractère	181
8.3	Programmation d'une communication en mode caractère	197
8.4	Mise au point d'une communication en mode caractère	212

Sous-chapitre 8.1

Généralités

A propos de la communication en mode caractère

Introduction

La communication en Mode caractère permet de réaliser des fonctions de dialogue et de communication entre les automates et des équipements parmi les systèmes suivants :

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

Sous-chapitre 8.2

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 de la communication en mode caractère du module BMX NOM 0200 dans un rack local	182
Ecran de configuration de la communication en mode caractère du module BMX NOM 0200 dans une station X80	185
Fonctions accessibles en mode caractère	188
Valeurs par défaut des paramètres de communication en mode caractère	189
Paramètre de détection de fin de message en mode caractère	190
Paramètres liés à la transmission en mode caractère	192
Paramètres liés aux signaux et à la ligne physique en mode caractère	194

Ecran de configuration de la communication en mode caractère du module BMX NOM 0200 dans un rack local

Général

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

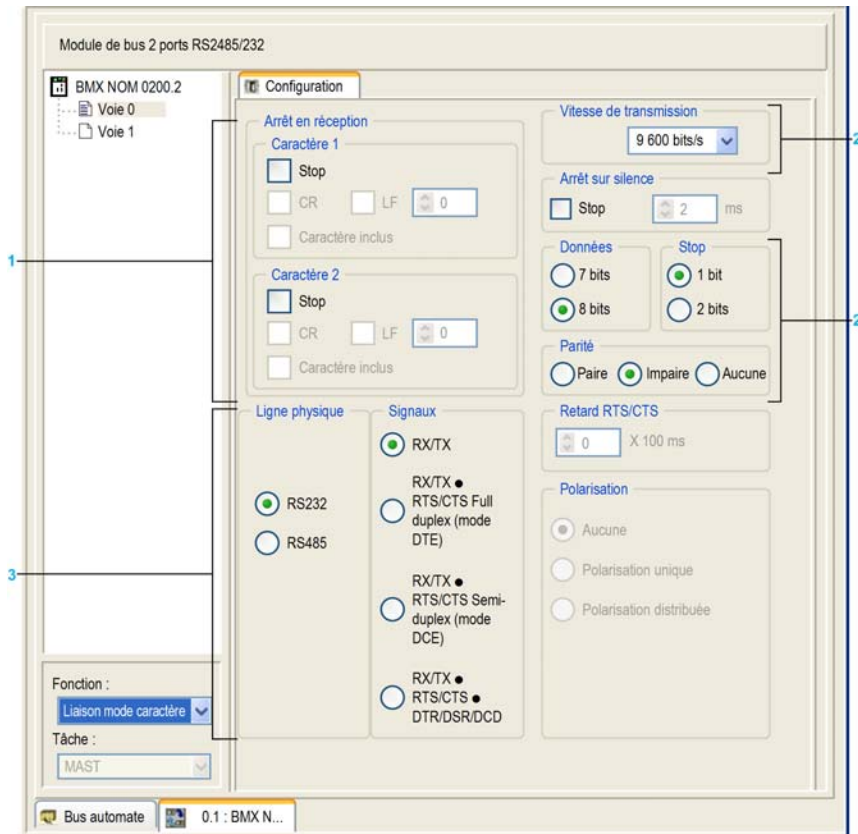
Accès à l'écran de configuration

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

Etape	Action
1	Ouvrez le sous-répertoire BMX NOM 0200 dans le navigateur de projet (<i>voir page 136</i>).
2	Sélectionnez la voie à configurer et la fonction de liaison en mode caractère dans l'écran qui apparaît.

Ecran de configuration en mode caractère

La figure ci-dessous représente un écran de configuration par défaut d'une communication en mode caractère sur la voie 0 :



Description

Ces zones permettent de configurer les paramètres de voie. En mode connecté, ces zones sont accessibles. En mode local, elles sont accessibles, mais certains paramètres ne le sont pas. Ils sont alors grisés.

Le tableau ci-dessous présente les différentes zones de l'écran de configuration de liaison de communication en mode caractère :

Touche	Élément	Commentaire
1	les paramètres concernant la détection de fin de message, (voir page 190)	Ces paramètres sont accessibles à partir de 2 zones : ● Arrêt en réception, ● Arrêt sur silence.
2	les paramètres concernant la transmission, (voir page 192)	Ces paramètres sont accessibles à partir de 4 zones : ● Vitesse de transmission, ● Données, ● Bits d'arrêt, ● Parité.
3	les paramètres concernant les signaux et la ligne physique. (voir page 194)	Ces paramètres sont accessibles à partir de 4 zones : ● Ligne physique, ● Signaux, ● Retard RTS/CTS, ● Polarisation.

NOTE : Dans cet exemple les zones "Polarisation" et "Retard RTS/CTS" sont grisées respectivement parce qu'une ligne physique RS232 et des signaux RX/TX ont été choisis.

Ecran de configuration de la communication en mode caractère du module BMX NOM 0200 dans une station X80

Général

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

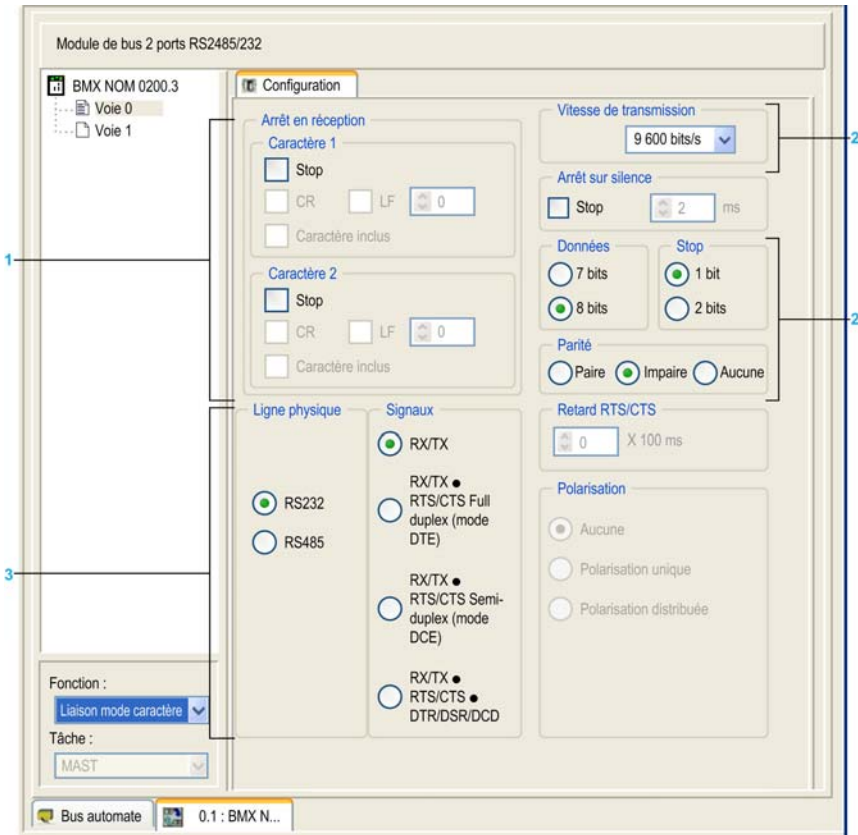
Accès à l'écran de configuration

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

Etape	Action
1	Ouvrez le sous-répertoire BMX NOM 0200 dans le navigateur de projet (<i>voir page 136</i>).
2	Sélectionnez la voie à configurer et la fonction de liaison en mode caractère dans l'écran qui apparaît.

Ecran de configuration en mode caractère

La figure ci-dessous représente un écran de configuration par défaut d'une communication en mode caractère sur la voie 0 :



Description

Ces zones permettent de configurer les paramètres de voie. En mode connecté, ces zones sont accessibles. En mode local, elles sont accessibles, mais certains paramètres ne le sont pas. Ils sont alors grisés.

Le tableau ci-dessous présente les différentes zones de l'écran de configuration de liaison de communication en mode caractère :

Touche	Elément	Commentaire
1	les paramètres concernant la détection de fin de message, (voir page 190)	Ces paramètres sont accessibles à partir de 2 zones : ● Arrêt en réception, ● Arrêt sur silence.
2	les paramètres concernant la transmission, (voir page 192)	Ces paramètres sont accessibles à partir de 4 zones : ● Vitesse de transmission, ● Données, ● Bits d'arrêt, ● Parité.
3	les paramètres concernant les signaux et la ligne physique. (voir page 194)	Ces paramètres sont accessibles à partir de 4 zones : ● Ligne physique, ● Signaux, ● Retard RTS/CTS, ● Polarisation.

NOTE : Dans cet exemple les zones "Polarisation" et "Retard RTS/CTS" sont grisées respectivement parce qu'une ligne physique RS232 et des signaux RX/TX ont été choisis.

Fonctions accessibles en mode caractère

Présentation

L'accessibilité des fonctions pour la configuration d'une liaison série d'un BMX NOM 0200 utilisant le protocole mode caractère est fonction de la ligne physique utilisée.

Fonctions accessibles

Le tableau ci-dessous indique les différentes fonctions configurables selon le type de liaison série utilisée :

Fonction	Liaison RS 485 (Voie 0 ou Voie 1)	Liaison RS 232 (Voie 0)
Vitesse de transmission	X	X
Données	● 7 bits ● 8 bits	● 7 bits ● 8 bits
Arrêter	● 1 bit ● 2 bits	● 1 bit ● 2 bits
Parité	● Impaire ● Paire ● Rien	● Impaire ● Paire ● Rien
Arrêt en réception	X	X
Arrêt sur silence	X	X
Signaux RX/TX	X	X
Signaux RTS/CTS	-	X
Retard RTS/CTS	-	X
Signaux DTR/DSR/DCD	-	X
Polarisation	X	-

X Fonction accessible

- Fonction inaccessible

Valeurs par défaut des paramètres de communication en mode caractère

Présentation

Les paramètres d'une communication en Mode caractère possèdent tous des valeurs par défaut.

Valeurs par défaut

Le tableau suivant présente les valeurs par défaut des paramètres d'une communication en mode caractère sur Voie 0 et Voie 1 du module BMX NOM 0200 :

Paramètre de configuration	Valeur sur la voie 0	Valeur sur la voie 1
Tracé physique	RS232	RS485
Signaux	RX/TX	RX/TX (valeur unique)
Vitesse de transmission	9 600 bits/s	9 600 bits/s
Parité	Impaire	Impaire
Bits de données	8 bits	8 bits
Bits d'arrêt	1 bit	1 bit
Polarisation	Rien (valeur unique)	Rien

Paramètre de détection de fin de message en mode caractère

Présentation

Après avoir configuré la voie de communication, vous devez renseigner les paramètres dédiés aux détections de fins de messages.

Ces paramètres sont accessibles à partir de 2 zones :

- zone Arrêt en réception : condition d'arrêt par caractère spécial,
- zone Arrêt sur silence : condition d'arrêt par silence.

Conditions d'utilisation

La sélection de l'arrêt sur silence entraîne la désélection de l'arrêt par caractère et inversement.

Zone Arrêt en réception

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

Arrêt en réception

Caractère 1

☒ Arrêter

☐ CR ☒ LF : 10

☐ Caractères inclus

Caractère 2

☒ Arrêter

☒ CR ☒ LF : 13

☐ Caractères inclus

Une requête de réception peut se terminer dès la réception d'un caractère particulier.

En cochant l'option Arrêt, il est possible d'activer et de paramétrer l'arrêt en réception par un caractère de fin :

- CR : permet de détecter la fin du message par un retour chariot,
- LF : permet de détecter la fin de message par un retour à la ligne,
- champ de saisie : permet d'identifier un caractère de fin, différent du caractère CR ou LF, par une valeur décimale :
 - 0...255 si les données sont codées sur 8 bits,
 - 0...127 si les données sont codées sur 7 bits.
- caractère inclus : permet d'inclure le caractère de fin dans la table de réception de l'application automate.

Il est possible de configurer 2 caractères de fin de réception. Dans la fenêtre ci-dessus, la fin de réception d'un message est détectée par le caractère LF ou CR.

Zone Arrêt sur silence

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

A screenshot of a configuration window titled 'Arrêt sur silence'. It contains a checked checkbox labeled 'Arrêter' and a text input field with the value '1' followed by the unit 'ms'.

Cette zone permet de détecter la fin d'un message en réception par absence de caractère de fin pendant un temps donné.

L'arrêt sur silence est valide lorsque la case Arrêt est cochée. La durée du silence, exprimée en millisecondes, est fixée par le champ de saisie.

NOTE : Les valeurs disponibles sont comprises entre 1 ms et 10 000 ms et dépendent de la vitesse de transmission choisie.

Paramètres liés à la transmission en mode caractère

Présentation

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

Ces paramètres sont accessibles à partir de 4 zones :

- Zone Vitesse de transmission,
- Zone Données,
- Zone Stop,
- Zone Parité.

Zone Vitesse de transmission

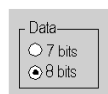
Cette zone de configuration apparaît dans l'écran comme représenté ci-dessous :



Cette zone permet de sélectionner la vitesse de transmission du protocole Mode caractère. Cette vitesse est en conformité avec les autres équipements. Les valeurs configurables sont 300, 600, 1200, 2400, 4800, 9600, 19200, 57600 et 115200 bits/s (seulement sur la voie 0 en mode RS232).

Zone Données

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



Cette zone permet de renseigner la taille des données échangées sur la liaison.

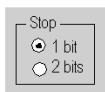
Les valeurs disponibles sont :

- 7 bits
- 8 bits

Il est conseillé d'ajuster le nombre de bits de données en fonction des équipements distants.

Zone Stop

La zone se présente comme ci-dessous :



Cette zone permet de renseigner le nombre de bits d'arrêt utilisés pour communiquer. Il est conseillé d'ajuster le nombre de bits d'arrêt en fonction des équipements distants.

Les valeurs configurables sont :

- 1 bit
- 2 bits

Zone Parité

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



Cette zone permet de définir l'adjonction ou non d'un bit de parité ainsi que son type. Il est conseillé d'ajuster la parité en fonction des équipements distants.

Les valeurs configurables sont :

- Paire
- Impaire
- Rien

Paramètres liés aux signaux et à la ligne physique en mode caractère

Vue d'ensemble

Après avoir configuré la voie de communication, vous devez renseigner les paramètres de la ligne physique et des signaux.

Ces paramètres sont accessibles à partir de 3 zones :

- zone Ligne Physique,
- zone Signaux,
- zone Retard RTS/CTS.

Zone Ligne Physique

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

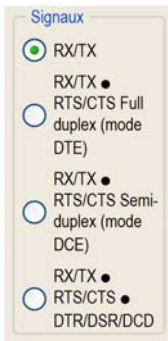


Cette zone permet de choisir le type de la ligne physique du port série du module BMX NOM 0200 parmi les deux types suivants :

- ligne RS 232,
- ligne RS 485,

Zone Signaux

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



Cette zone permet de sélectionner les signaux supportés par la ligne physique RS 232 :

- RX/TX
- RX/TX + RTS/CTS Full Duplex (mode DTE)
- RX/TX + RTS/CTS Semi-Duplex (mode DCE)
- RX/TX + RTS/CTS + DTR/DSR/DCD

Si la ligne configurée est de type RS 485, la zone est entièrement grisée et la valeur par défaut est RX/TX.

Zone retard RTS/CTS

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



La zone de temporisation RTS/CTS n'est disponible que quand les deux cases RS232 et RX/TX+RTS/CTS ou RX/TX+RTS/CTS+DTR/DSR/DCD sont cochées. Un contrôle de flux matériel RTS/CTS est utilisé.

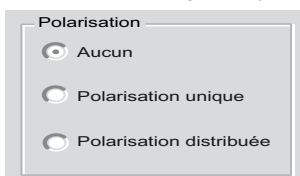
L'algorithme de contrôle de flux matériel RTS/CTS vise à éviter le débordement de tampon de réception (full duplex).

La temporisation RTS/CTS correspond au délai d'attente entre la montée de RTS et la montée de CTS. Une valeur de temporisation RTS/CTS différente de 0 correspond aussi au temps d'attente maximum entre chaque transmission de caractère après la montée des signaux RTS et CTS. Si la valeur est réglée sur 0, les puces UART peuvent rester coincées en état d'attente pendant un temps infini jusqu'à la montée de CTS, donc la valeur 0 n'est utilisée que dans des cas particuliers tels que bouclage du signal RTS sur le signal CTS de façon à vérifier qu'aucun câble de connexion n'est défectueux.

NOTE : la valeur par défaut est 0 ms.

Zone Polarisation

Cette zone de configuration présentée ci-dessous est accessible quand « RS485 » est sélectionné dans la zone « Ligne physique » :



Cette zone offre la possibilité de choix entre trois types de configuration pour la polarisation sur la voie :

- **Rien** pour n'utiliser aucune polarisation au cas où vous disposez de votre propre terminaison.
- **Polarisation unique** pour utiliser une polarisation à faible impédance telle que sur les réseaux Modbus (l'objectif de ce type de polarisation est de laisser le maître maintenir l'état par défaut).
- **Polarisation distribuée** pour utiliser une polarisation à haute impédance (l'objectif de ce type de polarisation est de laisser chaque appareil contribuer au maintien de l'état par défaut).

Sous-chapitre 8.3

Programmation d'une communication en mode caractère

Objet de cette section

Cette partie décrit l'aspect programmation 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
Fonctions de communication en mode caractère	198
Informations sur le mode expert	207

Fonctions de communication en 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 envoie une chaîne de caractères d'une longueur maximum de 16 x 1024 octets.
- INPUT_CHAR lit une chaîne de caractères d'une longueur maximum de 16 x 1024 octets.
- INPUT_BYTE lit des tableaux d'octets d'une longueur maximum de 16 x 1024 octets.

Le BMX NOM 0200 peut stocker jusqu'à 16 trames en émission ou en réception. Les trames dans les tampons sont gérées selon leur ordre d'arrivée. Sur les lignes RS-232, elles sont gérées en mode Full Duplex.

NOTE : pour confirmer le bit d'activité de la fonction INPUT_CHAR, il est nécessaire de définir une valeur de timeout si la voie est configurée sans **arrêt en cas de silence**.

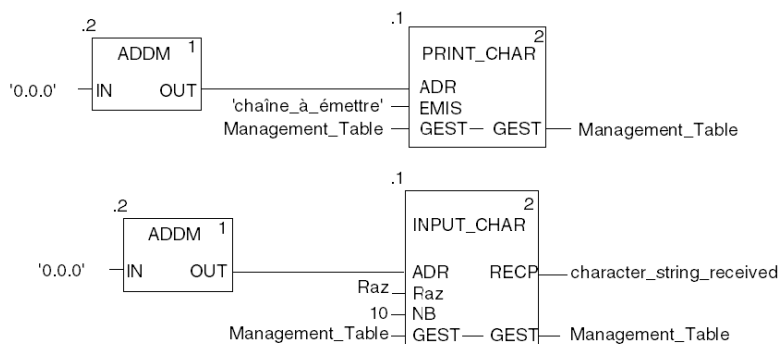
Nombre de trames reçues dans les tampons

Lorsque le port Modbus est configuré en mode caractère, %MWr.m.c.7 indique le nombre de trames dans le tampon de réception du BMX NOM 0200.

Ce type WORD est incrémenté chaque fois que le BMX NOM 0200 a reçu une trame sur une ligne RS-232.

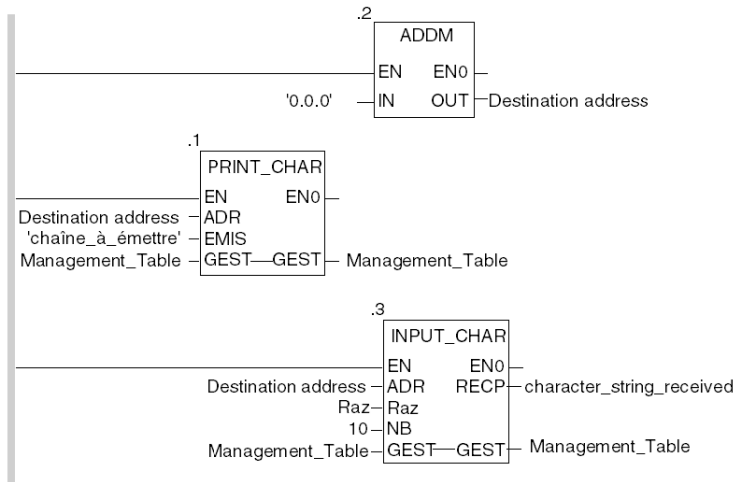
Exemple de programmation en FBD

La figure suivante représente un exemple de programmation en langage FBD des fonctions de communication PRINT_CHAR et INPUT_CHAR :



Exemple de programmation en Ladder

La figure suivante représente un exemple de programmation en langage Ladder des fonctions de communication PRINT_CHAR et INPUT_CHAR :



Exemple de programmation en ST

Les lignes de code suivantes représentent un exemple de programmation des fonctions de communication PRINT_CHAR et INPUT_CHAR en langage ST :

```
PRINT_CHAR(ADDM('0.1.0'), 'string_to_send', Management_Table);  
INPUT_CHAR(ADDM('0.1.0'), reset_integer_to_0, 10, Management_Table,  
character_string_received);
```

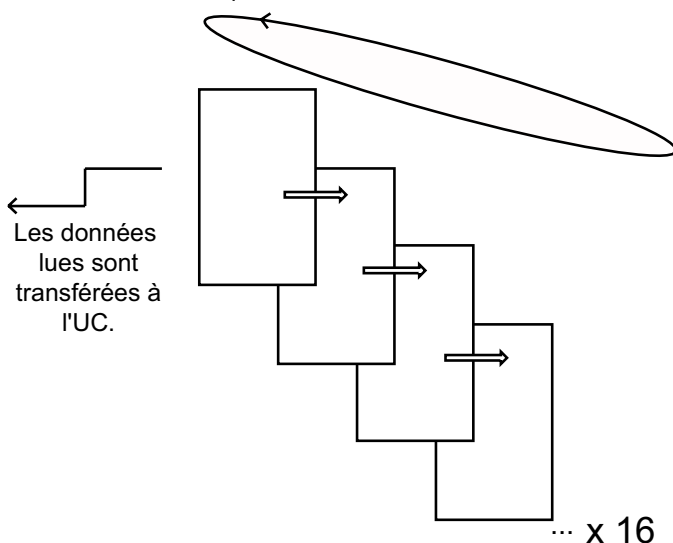
Caractéristiques de la fonction INPUT_CHAR

Si le paramètre d'entrée Reset a la valeur 1, tous les tampons sont d'abord réinitialisés, puis le module attend la réception de données. L'utilisation de cette option est conseillée pour démarrer correctement une réception en supprimant les anciennes données des tampons.

Mécanisme interne du module BMX NOM 0200

Les données reçues sont enregistrées dans 16 tampons cycliques en série de 1024 octets chacun.

Ce mécanisme est représenté ci-dessous :



Réception de données avec **INPUT_CHAR** ou **INPUT_BYTE**

Les trames sont récupérées par le programme d'application à l'aide de la fonction élémentaire (EF) **INPUT_CHAR** pour la réception d'une chaîne de caractères ou à l'aide de l'EF **INPUT_BYTE** pour la réception de données binaires.

L'EF **INPUT_CHAR** ou **INPUT_BYTE** peut être exécutée avant la réception des données par le module. Dans ce cas, le module attend les données de la ligne, puis les envoie à l'UC.

L'EF peut également être exécutée une fois la trame reçue (par exemple, après la vérification de %MWr.m.c.7 avec **READ_STS**). Dans ce cas, le module envoie immédiatement la trame en mémoire tampon à l'UC.

Il est également possible de forcer le module à attendre les données de la ligne en réglant le paramètre **Reset** de l'EF sur 1. Dans ce cas, les données précédemment placées en mémoire tampon sont vidées et le **BMX NOM 0200** attend de nouvelles données à envoyer à l'UC.

Mode Réception

Le module **BMX NOM 0200** peut stocker jusqu'à 16 trames.

Peu importe qu'un critère de fin de trame ait été défini ou non, le **BMX NOM 0200** peut être configuré en interne en mode Message ou en mode Brut :

- Le mode Message est activé lorsqu'une condition de fin de trame a été spécifiée.
Dans ce mode, le paramètre **NB** de **INPUT_CHAR** ou **INPUT_BYTE** définit le nombre d'octets envoyés à l'UC.
Les valeurs spéciales **NB = 0** et **NB = 1024** spécifient l'envoi de la trame entière à l'UC.

-
- Les autres valeurs de NB spécifient le nombre d'octets à envoyer à l'UC. Si davantage d'octets ont été reçus avant l'appel de INPUT_CHAR ou INPUT_BYTE, les octets restants sont rejetés.
- Le mode Brut est activé lorsqu'aucune condition de fin de trame n'a été spécifiée. Le mode caractère est un protocole "orienté trame". En réception, une fin de trame est détectée par le premier des événements suivants :
 - une durée de silence spécifiée dans la zone **Arrêt sur silence** de l'écran de configuration des communications en mode caractère (*voir page 182*)
 - la réception du caractère 1 ou 2 spécifié dans les zones **Arrêt à réception** de l'écran de configuration des communications en mode caractère (*voir page 182*)
 - la réception de plus de 1024 octets

Lorsqu'une fin de trame est détectée, le compteur interne %MW.r.m.c.7 est incrémenté.

Taille maximale des données

La taille maximale d'une trame envoyée par le module BMX NOM 0200 à l'UC est de 1024 octets. Cependant, en interne, la trame de réception a une taille maximale de 1025 octets si un octet de fin de trame est configuré. Cet octet n'est pas inclus dans les données envoyées à l'UC.

Trames de taille nulle

Les trames de taille nulle sont rejetées. Si un octet de fin de trame est configuré et non demandé dans les données, une trame de taille nulle reçue par le module BMX NOM 0200 n'est pas envoyée à l'UC. Dans ce cas, si un octet de fin de trame est reçu sans donnée le précédant, cette trame est rejetée et aucune information n'est envoyée à l'UC.

Réception de plusieurs trames pendant une tâche MAST

Plusieurs trames peuvent être transmises par le module BMX NOM 0200 à l'UC pendant une tâche MAST, et plusieurs EF INPUT_CHAR peuvent être lancées en parallèle en faisant référence au même module BMX NOM 0200. Ceci peut être requis si une quantité importante de données arrive sur la ligne série.

Cancel et Timeout

Cancel et Timeout sont transmis au module BMX NOM 0200. La condition Timeout et les ordres d'annulation (Cancel) appliqués à une instance de INPUT_CHAR sont transmis au module BMX NOM 0200. La tâche en attente correspondante est supprimée de la file d'attente des tâches du module BMX NOM 0200.

Mécanisme interne du module BMX NOM 0200 : émission

Utilisez l'EF PRINT_CHAR pour envoyer les données sur la ligne série du module BMX NOM 0200.

NOTE : Si plusieurs trames ont été envoyées (plusieurs instances de PRINT_CHAR appelées) et qu'un silence a été configuré, le module BMX NOM 0200 insère un temps de silence entre chaque trame.

Il est possible de lancer jusqu'à 16 requêtes PRINT_CHAR : elles sont envoyées en série avec un silence entre chaque PRINT_CHAR.

Annulation d'un échange

Deux types de programmation permettent d'annuler un échange effectué par les fonctions PRINT_CHAR et INPUT_CHAR. Ces deux programmations sont présentées en langage ST ci-dessous :

- Utilisation de la fonction CANCEL :

```
IF (%MW40.0) THEN
  %MW200 := SHR(%MW40, 8);
  CANCEL(%MW200, %MW185);
END_IF;
```

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction PRINT_CHAR et il est à 1 lorsque la fonction de communication est active. Si ce bit est à 1, le programme exécute les instructions suivantes :

- Il déplace les bits %MW40 d'un octet (8 bits) vers la droite et charge l'octet correspondant au numéro d'échange de la communication dans le mot %MW200.
- Il annule l'échange dont le numéro est contenu dans le mot %MW200 à l'aide de la fonction CANCEL.

- Utilisation du bit d'annulation de la fonction de communication :

```
IF (%MW40.0) THEN
  SET(%MW40.1);
  PRINT_CHAR(ADDM('0.1.0'), 'chaîne_à_émettre', %MW40:4);
END_IF;
```

%MW40 correspond au paramètre GEST (table de gestion). %MW40.0 correspond au bit d'activité de la fonction PRINT_CHAR et il est à 1 lorsque la fonction de communication est active. Si ce bit est à 1, le programme met à 1 le bit %MW40.1 d'annulation de la fonction. Ceci a pour effet d'arrêter la communication de la fonction PRINT_CHAR.

NOTE : Avec l'utilisation du bit d'annulation de la fonction de communication, la fonction doit être appelée pour activer le bit d'annulation contenu dans le mot de gestion d'échanges de la fonction (%MW40 pour cet exemple).

NOTE : Lors de l'utilisation du bit d'annulation de la fonction de communication, il est possible d'annuler une communication à partir d'une table d'animation. Pour cela, il suffit de mettre à 1 le bit d'annulation de la fonction (%MW40.1 dans l'exemple).

NOTE : La fonction CANCEL utilise un mot de compte rendu pour la fonction CANCEL (%MW185 pour cet exemple).

NOTE : Cet exemple de programmation concerne la fonction PRINT_CHAR, mais il est également applicable à la fonction INPUT_CHAR.

Description des paramètres de la fonction ADDM

Le tableau suivant décrit les différents paramètres de la fonction ADDM :

Paramètre	Type	Description
IN	STRING	Adresse d'un équipement sur un bus ou sur une liaison série. La syntaxe de l'adresse est du type "r.m.c.node". L'adresse est composée des paramètres suivants : ● r : numéro du rack du système destinataire, toujours égal à 0 ● m : numéro d'emplacement du système destinataire dans le rack, toujours égal à 0 ● c : numéro de voie, toujours égal à 0 car la liaison série d'un système distant est toujours la voie 0 ● node : champ facultatif pouvant être SYS ou vide
OUT	ARRAY [0..7] OF INT	Tableau représentant l'adresse d'un équipement. Ce paramètre est utilisable par plusieurs fonctions de communication comme paramètre d'entrée.

Description des paramètres de la fonction PRINT_CHAR

Le tableau suivant décrit les différents paramètres de la fonction PRINT_CHAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse du message recevant la voie en mode caractère donnée par le paramètre OUT de la fonction ADDM.
EMIS	STRING	Chaîne de caractères à émettre.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● Mot de rang 1 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : numéro d'échange, ○ octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1) ● Mot de rang 2 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : compte rendu d'opération ○ octet de poids faible : compte rendu de communication ● Mot de rang 3 : mot géré par l'utilisateur et qui définit le temps de réponse maximum en utilisant une base de temps de 100 ms. ● Mot de rang 4 : mot géré par l'utilisateur et qui définit la longueur de l'échange : ○ Si cette longueur de paramètre est définie à 0, le système envoie la chaîne entière. ○ Si cette longueur de paramètre est supérieure à la longueur de la chaîne, l'erreur 16#0A (Taille insuffisante du tampon d'envoi) est renvoyée dans le 2ème mot de gestion et aucun caractère n'est envoyé.

Description des paramètres de la fonction INPUT_CHAR

Le tableau suivant décrit les différents paramètres de la fonction INPUT_CHAR :

Paramètre	Type	Description
ADR	ARRAY [0..7] OF INT	Adresse du message recevant la voie en mode caractère donnée par le paramètre OUT de la fonction ADDM.
RAZ	INT	Réinitialisation. Ce paramètre permet de réinitialiser la mémoire de réception du coupleur : ● Valeur = 0 : pas de réinitialisation de la mémoire ● Valeur = 1 : réinitialisation de la mémoire NOTE : %MW.r.m.7 est remis à 0.
NB	INT	Longueur de la chaîne de caractères à recevoir.

Paramètre	Type	Description
GEST	ARRAY [0..3] OF INT	Table de gestion de l'échange composée des mots suivants : ● Mot de rang 1 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : numéro d'échange, ○ octet de poids faible : bit d'activité (rang 0) et bit d'annulation (rang 1) ● Mot de rang 2 : mot géré par le système et composé de 2 octets : ○ octet de poids fort : compte rendu d'opération ○ octet de poids faible : compte rendu de communication ● Mot de rang 3 : mot géré par l'utilisateur et qui définit le temps de réponse maximum en utilisant une base de temps de 100 ms. ● Mot de rang 4 : mot géré par le système et qui définit la longueur de l'échange.
RECP	STRING	Chaîne de caractères reçue. Cette chaîne est mémorisée dans une chaîne de caractères.

Description des paramètres de la fonction INPUT_BYTE

Le tableau suivant décrit les différents paramètres de la fonction INPUT_BYTE :

Paramètre	Type	Commentaire
ADR	ARRAY [0..7] OF INT	Pour les automates Modicon M340 : ● L'adresse de la voie en mode caractère de réception du message est indiquée par la fonction ADDM. ● La syntaxe de l'adresse est de type ADDM (' r . m . c . node '). Le champ "node" est facultatif et il peut être SYS ou vide (par exemple, ADDM(' 0 . 0 . 0 . SYS ') est égal à ADDM(' 0 . 0 . 0 ').
RAZ	INT	Réinitialisation. Ce paramètre permet de réinitialiser la mémoire de réception du coupleur. ● valeur 0 : pas de réinitialisation de la mémoire ● valeur 1 : réinitialisation de la mémoire NOTE : %MW.r.m.7 est remis à 0. REMARQUE : Sur les automates Modicon M340, l'EF INPUT_BYTE peut être programmée avec ou sans ce paramètre.

Paramètre	Type	Commentaire
NB	INT	Longueur du tampon ou nombre d'octets à recevoir. • Valeur 0 : message lu dès qu'il est disponible sur la voie. Ici, une condition d'arrêt doit être spécifiée dans l'écran de configuration. • Valeur supérieure à 0 : indique le nombre d'octets à lire. REMARQUE : Le caractère de fin de message par défaut est un retour chariot (CR).

Durée minimale du silence

Lorsque la case Stop est cochée, un silence sur la ligne d'entrée est l'une des conditions qui déterminent la détection d'une fin de trame.

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.

Informations sur le mode expert

Communication en mode expert

Le mode expert est un ensemble de commandes qui peut être envoyé au module pour obtenir des fonctionnalités supplémentaires.

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr.m.c.24	CONTROL	Explicite	INT	Signal de commande, changer de protocole
%MWr.m.c.24.0		Explicite	BOOL	Effacer les compteurs locaux
%MWr.m.c.24.4		Explicite	BOOL	Modifier les synchronisations internes de silence (%MW30). La mise à jour de cette valeur risque de perturber le module si celui-ci est en cours de fonctionnement.
%MWr.m.c.24.5		Explicite	BOOL	Modifier l'octet de fin de trame en mode caractère 0 (%MW26) et l'octet 1 (%MW27)

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr.m.c.24.6		Explicite	BOOL	Modifier le mode de gestion du modem : HALF/FULL DUPLEX ● Si le bit est défini en même temps que RTS_ON (%MWr.m.c.24.10) fonctionne aussi avec RTS_OFF %MWr.m.c.24.11 et utilise DTR si .8 ou .9 est utilisé), le mode de modem half duplex est activé. ● Si ce bit est défini mais qu'aucun bit RTS ou DTR (%MWr.m.c.24.8, %MWr.m.c.24.9, %MWr.m.c.24.10, %MWr.m.c.24.11) ne l'est, le mode full duplex est activé. Le bit %MW26 sert à définir le délai de début (StartDelay) et le bit %MW27 le délai de fin (EndDelay). Les bits %MW24.5, %MW24.1 et %MW24.2 ne peuvent donc pas être utilisés simultanément. REMARQUE : l'utilisateur peut être amené à rétablir l'état correct des signaux RTS/DTR après que la commande a été acceptée.
%MWr.m.c.24.7		Explicite	BOOL	Enregistrer l'adresse esclave Modbus dans la mémoire FLASH (%MW25).
%MWr.m.c.24.8	DTR_ON	Explicite	BOOL	Définir le signal DTR (tension positive)
%MWr.m.c.24.9	DTR_OFF	Explicite	BOOL	Réinitialiser le signal DTR (tension négative)
%MWr.m.c.24.10		Explicite	BOOL	Définir le signal RTS (tension positive)

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr . m . c . 24 . 11		Explicite	BOOL	Réinitialiser le signal RTS (tension négative)
%MWr . m . c . 24 . 12	TO_MODBUS_MASTER	Explicite	BOOL	Basculer en mode maître
%MWr . m . c . 24 . 13	TO_MODBUS_SLAVE	Explicite	BOOL	Basculer en mode esclave
%MWr . m . c . 24 . 14	TO_CHAR_MODE	Explicite	BOOL	Basculer en mode caractère
%MWr . m . c . 25		Explicite	INT	Adresse esclave Modbus pour le stockage dans la mémoire FLASH
%MWr . m . c . 26		Explicite	INT	Nouvel octet de fin de trame (EOF) en mode caractère (eq %KW6) si le bit %MW24.5 est défini : ● Bit 0: 1 byte 1 is set below, 0 no more byte 1 ● Bit 1 : 1 ajoute l'octet 1, 0 n'ajoute pas l'octet 1 ● Bits 2 à 7 : doivent être de type null. Octet de poids fort : l'octet de fin de trame 1 Délai de début (StartDelay) si le bit %MW26 . 6 est défini. Délai à respecter lorsque le CTS est OK avant de commencer à envoyer la trame. Ce bit est utile pour les modems qui demandent un délai supplémentaire après le CTS ou qui ne gèrent pas le signal CTS (dans ce cas, le RTS doit être connecté au CTS). Ce délai est exprimé en millisecondes, avec une précision de 3 ms environ. Il ne peut être défini qu'en mode RS232.

Adresse	Symbole standard	Type d'échange	Type	Signification
%MWr . m . c . 27		Explicite	INT	Nouvel octet de fin de trame (EOF) en mode caractère (eq %KW7) si le bit %MW24 . 5 est défini : ● Bit 0: 1 byte 2 is set below, 0 no more byte 2 ● Bit 1 : 1 ajoute l'octet 2, 0 n'ajoute pas l'octet 2 ● Bits 2 à 7 : doivent être de type null. Octet de poids fort : l'octet de fin de trame 2 Délai de début (StartDelay) si le bit %MW24 . 6 est défini. Délai à respecter après l'envoi d'une trame, avant d'émettre le signal RTS de façon à donner suffisamment de temps au MODEM pour qu'il envoie la trame avant la fin de communication. Ce délai est exprimé en millisecondes, avec une précision de 3 ms environ. Il ne peut être défini qu'en mode RS232.
%MWr . m . c . 28		Explicite	INT	Réservé
%MWr . m . c . 29		Explicite	INT	Réservé
%MWr . m . c . 30		Explicite	INT	silence : délai intertrames en millisecondes [0 à 10000]. La valeur utilisée dépend de la vitesse. Si elle est inférieure ou supérieure aux valeurs possibles, la limite inférieure ou supérieure est appliquée, et la commande est acceptée. La valeur 0 signifie « aucun silence ».
%MWr . m . c . 31		Explicite	INT	Réservé
%MWr . m . c . 32		Explicite	INT	Réservé

Exemple de code

```
if HalfModemChar then
  HalfModemChar:=false;
  %MW0.9.0.24:=16#0440;(* basculer en mode half duplex avec RTS *)
  %MW0.9.0.26:=12;(* 12 ms à attendre avant de procéder à l'envoi quand
CTS est activé *)
  %MW0.9.0.27:=9; (* laisser RTS activé pendant 9 ms après la fin de
l'envoi *)
  write_cmd(%ch0.9.0);(* envoyer une commande et des données à la voie NOM
*)
end_if;
```

Sous-chapitre 8.4

Mise au point d'une communication en mode caractère

Ecran de mise au point d'une communication en mode caractère

Général

L'écran de mise au point d'une communication en 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 d'une communication en Mode caractère :

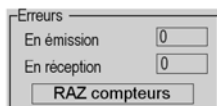
Etape	Action
1	Accédez à l'écran de configuration d'une communication en Mode caractère. (voir page 182)
2	Sélectionnez l'onglet "Mise au point" dans l'écran qui apparaît.

Description de l'écran de mise au point

L'écran de mise au point est composé d'une zone Erreurs et une zone Signaux (si RS232).

Zone Erreurs

La zone Erreurs se présente comme suit :



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

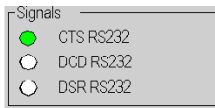
- **en transmission** : correspond au nombre d'interruptions en émission (image du mot %MW4),
- **en réception** : correspond au nombre d'interruptions en réception (image du mot %MW5),

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

Zone Signaux

Cette zone ne s'affiche que si RS232 est sélectionné dans l'écran de configuration. Si RS485 est sélectionné dans l'écran de configuration, cette fenêtre n'apparaît pas du tout.

La zone Signaux se présente comme suit :



Cette zone indique l'activité des signaux :

- **CTS RS232** : indique l'activité du signal CTS.
- **DCD RS232** : indique l'activité du signal DCD.
- **DSR RS232** : indique l'activité du signal DSR.

Chapitre 9

Diagnostics de module BMX NOM 0200

Sous-chapitre 9.1

Diagnostics de module BMX NOM 0200

Objet de cette section

Cette section décrit l'aspect de diagnostics de la mise en œuvre d'un module de communication BMX NOM 0200.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Diagnostics du module BMX NOM 0200	217
Diagnostic détaillé par voie de communication	219

Diagnostics du module BMX NOM 0200

Vue d'ensemble

La fonction Diagnostic du module affiche, le cas échéant, les anomalies en cours, classées par catégorie :

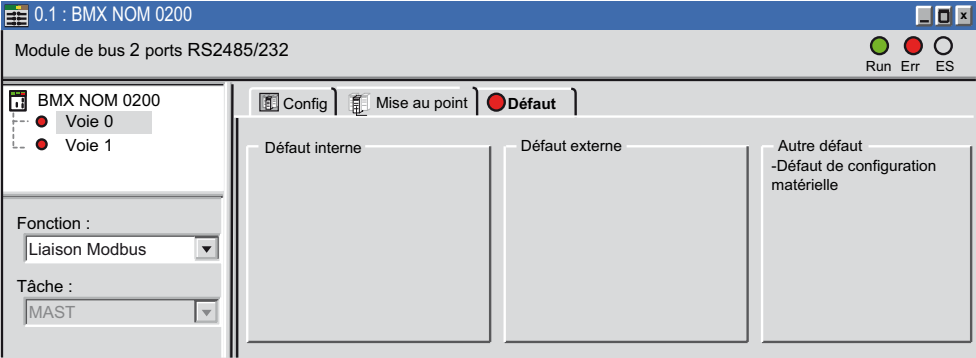
- **Erreur interne détectée :**
 - événement sur le module
- **Événement externe :**
 - contrôle de l'écriture (câble rompu, surcharge ou court-circuit),
- **Autres anomalies :**
 - voie inutilisable,
 - anomalie de configuration,
 - module absent ou hors tension.

Un module sur lequel un défaut est détecté est signalé par le passage au rouge d'un certain nombre de voyants, tels que :

- dans l'éditeur de configuration niveau rack :
 - le voyant du numéro du rack,
 - le voyant du numéro d'emplacement du module sur le rack.
- dans l'éditeur de configuration niveau module :
 - les voyants **Err** et **I/O**, selon le type d'erreur,
 - le voyant **Voie** dans la zone **Voie**.

Accès à l'écran de diagnostic du module

Le tableau ci-dessous donne la marche à suivre pour accéder à l'écran de diagnostic du module.

Etape	Action
1	Accédez à l'écran de mise au point du module.
2	Cliquez sur la référence du module dans la zone de la voie et sélectionnez l'onglet Défaut. Résultat : la liste des erreurs détectées du module apparaît.  Remarque : il n'est pas possible d'accéder à l'écran de diagnostic du module lorsqu'une erreur de configuration, une panne majeure ou l'absence du module est détectée. Le message suivant apparaît alors sur l'écran : " « Le module est absent ou différent de celui configuré à cette position. »

Liste d'erreurs détectées sur le module

Le tableau récapitulatif ci-dessous présente les différentes erreurs détectées pour un module de communication :

Classement d'erreurs détectées	Les objet langage
Défaut interne : ● erreur détectée du module	● %MWr.m.MOD.2.0
Défaut externe : ● Bornier	● %MWr.m.MOD.2.2
Autre défaut : ● Voie(s) en défaut ● Défaut de configuration matérielle ● Module absent ou désactivé	● %MWr.m.MOD.2.1 ● %MWr.m.MOD.2.5 ● %MWr.m.MOD.2.6

Diagnostic détaillé par voie de communication

Vue d'ensemble

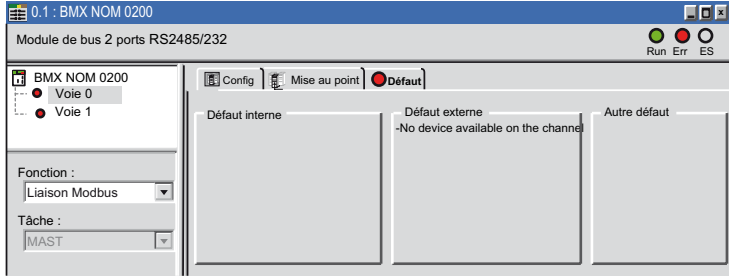
La fonction Diagnostic de la voie affiche, le cas échéant, les erreurs détectées en cours classées par catégorie :

- **Erreur interne détectée**
 - autotest en cours
- **Evénements externes**
 - appareil absent
 - équipement défaillant
 - dépassement de délai de communication de liaison série
- **Autres erreurs détectées**
 - erreur d'outil ligne
 - erreur de configuration
 - perte de communication
 - erreur d'application

Une erreur détectée de voie est indiquée dans l'onglet **Mise au point** par le passage au rouge du voyant  situé dans la colonne **Erreur**.

Accès à l'écran de diagnostic de voie

Le tableau ci-dessous donne la marche à suivre pour accéder à l'écran diagnostic de la voie.

Etape	Action
1	Accédez à l'écran de mise au point du module.
2	Pour la voie inutilisable, cliquez sur le bouton  situé dans la colonne Erreur. Résultat : la liste des erreurs de la voie apparaît.  Remarque : l'accès aux informations de diagnostic de la voie est également possible par programme (instruction READ_STS).

Liste d'erreurs détectées sur la voie

Le tableau récapitulatif ci-dessous présente les différentes erreurs détectées pour une liaison série configurée :

Classement d'erreurs détectées	Objets langage
Défaut interne : ● Autotest en cours	● %MWr.m.c.2.4
Défaut externe : ● Aucun appareil disponible sur la voie ● Défaut d'appareil ● Erreur de dépassement de délai (CTS)	● %MWr.m.c.2.0 ● %MWr.m.c.2.1 ● %MWr.m.c.2.3
Autre défaut : ● Erreur d'outil ligne ● Défaut de configuration matérielle ● Défaut de communication avec l'automate ● Erreur d'application	● %MWr.m.c.2.2 ● %MWr.m.c.2.5 ● %MWr.m.c.2.6 ● %MWr.m.c.2.7

Chapitre 10

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
10.1	Objets langage et IODDT des communications Modbus et Mode caractère	222
10.2	Objets langage et IODDT génériques pour les protocoles de communication	230
10.3	Objets langage et IODDT associés aux communications Modbus	234
10.4	Objets langage et IODDT associés aux communications Modbus en mode caractère	243
10.5	IODDT de type T_GEN_MOD applicable à tous les modules	253
10.6	Objets langage et DDT d'équipement associés aux communications Modbus	255

Sous-chapitre 10.1

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	223
Objets langage à échange implicite associés à la fonction métier	224
Objets langage à échanges explicites associés à la fonction métier	225
Gestion des échanges et des comptes rendus avec des objets explicites	227

Présentation des objets langage pour les communications Modbus et Mode caractère

Général

Les IODDT sont prédéfinis par le constructeur. Ils contiennent des objets langage d'entrées/sorties appartenant à la voie d'un module métier.

Les communications Modbus et Mode caractère ont 3 IODDT associés :

- T_COM_STS_GEN applicable à tous les protocoles de communication, sauf Fipio et Ethernet,
- T_COM_MB_BMX spécifique à la communication Modbus,
- T_COM_CHAR_BMX spécifique à la communication Mode caractère.

NOTE : Les variables IODDT peuvent être créées de deux façons :

- à l'aide de l'onglet Objets d'E/S (*voir Unity Pro, Modes de marche*),
- à l'aide de l'éditeur de données.

Types objets langage

Dans chacun des IODDT, se trouve un ensemble d'objets langage permettant de les commander et de vérifier leur 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 processeur,
- Objets à échanges explicites : ces objets sont échangés à la demande de l'application, en utilisant les instructions d'échanges explicites.

Les échanges implicites concernent les états des processeurs, des signaux de communication, des esclaves...

Les échanges explicites permettent de paramétrer le processeur et de le diagnostiquer.

Objets langage à échange implicite 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 aux 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 RUN ou en STOP.

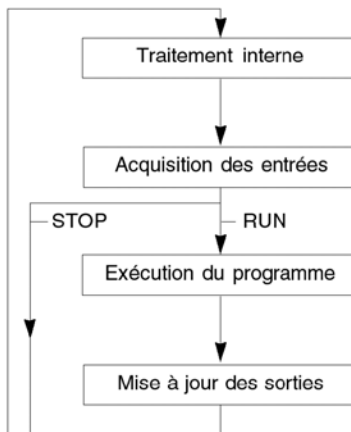
Les sorties (%Q et %QW) sont mises à jour en fin de tâche, uniquement lorsque l'automate est en RUN.

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 *Unity Pro, Gestion des E/S, Bibliothèque de blocs*) : lecture des mots d'état
- WRITE_CMD (voir *Unity Pro, Gestion des E/S, Bibliothèque de blocs*) : écriture des mots de commande

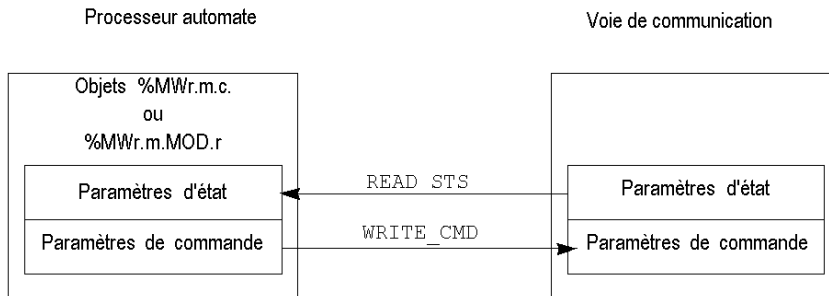
Ces échanges s'appliquent à un ensemble d'objets %MW de même type (état, commande ou paramètre) appartenant à une voie.

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 :



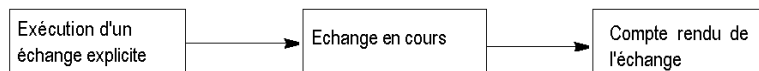
Gestion des échanges

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 Unity Pro, Gestion des E/S, Bibliothèque de blocs*) ;
- le compte rendu de l'échange (*voir Unity Pro, 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 (%MWrr . 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 (%MW_r.m.c.0) : échange en cours.
- EXCH_RPT (%MW_r.m.c.1) : compte rendu.

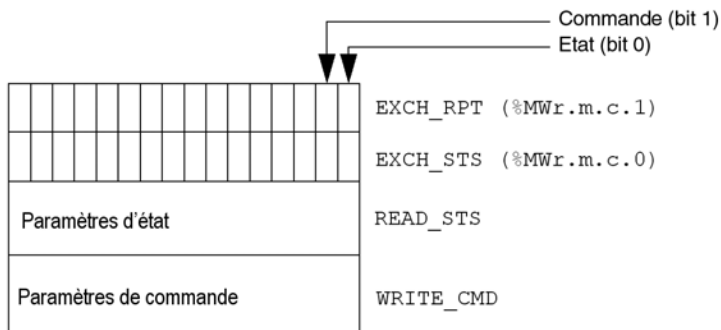
NOTE :

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 (%MWr . m . c . 0) et EXCH_RPT (%MWr . m . c . 1) est associé à un type de paramètre :

- Les bits de rang 0 sont associés aux paramètres d'état :
 - le bit STS_IN_PROGR (%MWr . m . c . 0 . 0) indique si une demande de lecture des mots d'état est en cours.
 - Le bit STS_ERR (%MWr . m . c . 1 . 0) 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 (%MWr . m . c . 0 . 1) indique si des paramètres de commande sont envoyés à la voie du module,
 - le bit CMD_ERR (%MWr . m . c . 1 . 1) 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 (%MWr . m . MOD . 0) et EXCH_RPT (%MWr . m . MOD . 1) 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 (%MWr . m . c . 0) :

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	R	Lecture des mots d'état de la voie en cours.	%MWr . m . c . 0 . 0
CMD_IN_PROGR	BOOL	R	Echange de paramètres de commande en cours	%MWr . m . c . 0 . 1
ADJ_IN_PROGR	BOOL	R	Echange de paramètres de réglage en cours	%MWr . m . c . 0 . 2
RECONF_IN_PROGR	BOOL	R	Reconfiguration du module en cours	%MWr . m . c . 0 . 15

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 (%MWr . m . c . 0 . 0) = 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 10.2

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 de l'IODDT de type T_COM_STS_GEN	231
Détails sur les objets à échange explicite de l'IODDT de type T_COM_STS_GEN	232

Informations détaillées sur les objets à échange implicite de l'IODDT de type T_COM_STS_GEN

Présentation

Le tableau suivant présente les objets à échange implicite de l'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 (%lr.m.c.ERR) :

Symbole standard	Type	Accès	Signification	Adresse
CH_ERROR	EBOOL	R	Bit erreur de la voie de communication.	%lr.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 (%MWr.m.c.0) :

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

Compte rendu d'échanges explicites : 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	R	Erreur détectée de lecture des mots d'état de la voie.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Erreur détectée lors d'un échange de paramètres de commande.	%MWr.m.c.1.1

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 10.3

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	235
Informations détaillées sur les objets à échanges implicites IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT	236
Informations détaillées sur les objets à échanges explicites IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT	237
Informations détaillées sur les objets langage associés au mode de communication Modbus	240

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

Informations détaillées sur les objets à échanges implicites 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 des types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT applicables aux communications série Modbus. Ils diffèrent au niveau de la **disponibilité des objets de configuration** (*voir page 239*).

Bit CH_ERROR

Le tableau suivant présente 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 détectée de la voie de communication.	%Ir.m.c.ERR

Objet mot en mode Modbus maître

Le tableau ci-dessous présente la signification du bit du mot INPUT_SIGNALS (%IW.r.m.c.0) :

Symbole standard	Type	Accès	Signification	Adresse
DCD	BOOL	R	Signal RS232 de détection de porteuse de données (applicable uniquement au module BMX NOM 0200)	%IW.r.m.c.0.0
CTS	BOOL	R	Prêt à envoyer le signal RS232	%IW.r.m.c.0.2
DSR	BOOL	R	Signal RS232 de jeu de données prêt (applicable uniquement au module BMX NOM 0200)	%IW.r.m.c.0.3

NOTE : Quand CTS est vert dans Punit, c'est que %IW.r.m.c.0.0 est à 1 et que la tension sur ce signal est positive. Cela vaut également pour DCD et DSR.

Objet mot en mode Modbus esclave

Les objets langage sont identiques à celle de la fonction Modbus Maître. Seuls les objets définis dans le tableau suivant diffèrent.

Le tableau ci-dessous présente la signification du bit du mot INPUT_SIGNALS (%IW.r.m.c.0) :

Symbole standard	Type	Accès	Signification	Adresse
LISTEN_ONLY	BOOL	R	Mode écoute seule :	%IW.r.m.c.0.8

Informations détaillées sur les objets à échanges explicites IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT

Vue d'ensemble

Cette partie présente les objets d'échange explicites des IODDT de types T_COM_MB_BMX et T_COM_MB_BMX_CONF_EXT applicables au protocole Modbus série et sont différents pour ce qui concerne la **disponibilité des objets de configuration** (*voir page 239*). 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.

Observations

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 (%MWr.m.c.0) :

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

Compte rendu d'échanges explicites : EXCH_RPT

Le tableau ci-dessous présente la signification des bits de compte rendu 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.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Anomalie lors d'un échange de paramètres de commande.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Anomalie lors de l'échange de paramètres de réglage en cours (non-application au module BMX NOM 0200).	%MWr.m.c.1.2

Défaillances détectées sur 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	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).

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#06 pour la fonction Modbus maître.	%MWr.m.c.3
PROTOCOL	INT	R	Octet 0 = 16#07 pour la fonction Modbus esclave.	%MWr.m.c.3

La lecture est effectuée par l'instruction READ_STS (IODDT_VAR1).

Commande 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 (Data Terminal Ready).	%MWr . m . c . 24 . 8
DTR_OFF	BOOL	L/E	Réinitialise le signal terminal de données prêt (Data Terminal Ready).	%MWr . m . c . 24 . 9
TO_MODBUS_MASTER	BOOL	L/E	Passe du mode caractère ou Modbus esclave au mode Modbus maître.	%MWr . m . c . 24 . 12
TO_MODBUS_SLAVE	BOOL	L/E	Passe du mode caractère ou Modbus maître au mode Modbus esclave.	%MWr . m . c . 24 . 13
TO_CHAR_MODE	BOOL	L/E	Passe 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, reportez-vous à la section relative aux **changements de protocole** (*voir page 262*).

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_CTRL_DATA	BOOL	L/E	Enregistre les données de commande dans la mémoire FLASH	%MWr . m . c . 24 . 7
CONTROL_DATA	BOOL	L/E	Adresse esclave Modbus à enregistrer dans la mémoire FLASH	%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 : ● La valeur 6 fait référence au maître. ● La valeur 7 fait référence à l'esclave.

Adresse	Type	Accès	Signification
%KWr.m.c.1	INT	R	L'octet 0 de ce mot correspond à la vitesse de transmission. Il peut avoir différentes valeurs : ● La valeur -2 (0xFE) correspond à 300 bits/s. ● La valeur -1 (0xFF) correspond à 600 bits/s. ● La valeur 0 (0x00) correspond à 1200 bits/s. ● La valeur 1 (0x01) correspond à 2400 bits/s. ● La valeur 2 (0x02) correspond à 4800 bits/s. ● La valeur 3 (0x03) correspond à 9600 bits/s. ● La valeur 4 (0x04) correspond à 19200 bits/s (valeur par défaut). ● La valeur 5 (0x05) correspond à 38400 bits/s. ● La valeur 6 (0x06) correspond à 57600 bits/s (applicable au module BMX NOM 0200 seulement) ● La valeur 7 (0x07) correspond à 115200 bits/s (applicable au module BMX NOM 0200 seulement) L'octet 1 de ce mot correspond au format : ● Bit 8 : nombre de bits (1 = 8 bits (RTU), 0 = 7 bits (ASCII)) ● bit 9 = 1 : gestion de la parité (1 = avec, 0 = sans) ● Bit 10 : type de parité (1 = impair, 0 = pair) ● Bit 11 : nombre de bits d'arrêt (1 = 1 bit, 0 = 2 bits) ● Bit 13 : ligne physique (1 = RS232, 0 = RS485) ● 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. ● 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).
%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.

Adresse	Type	Accès	Signification
%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 10.4

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.	244
Informations détaillées sur les objets à échange implicite IODDT de type T_COM_CHAR_BMX	245
Informations détaillées sur les objets à échange explicite IODDT de type T_COM_CHAR_BMX	246
Informations détaillées sur les objets langage associés à la configuration en mode caractère	249

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

Informations détaillées sur les objets à échange implicite IODDT de type T_COM_CHAR_BMX

Présentation

Les tableaux ci-dessous présentent les objets à échange implicite de l'IODDT de type T_COM_CHAR_BMX qui s'appliquent à la communication Mode caractère.

Bit d'erreur

Le tableau suivant présente la signification du bit d'erreur CH_ERROR (%Ir . m . c . ERR) :

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

Objet des signaux en entrée

Le tableau ci-dessous présente la signification du bit du mot INPUT_SIGNALS (%IWr . m . c . 0) :

Symbole standard	Type	Accès	Signification	Adresse
DCD	BOOL	R	Signal détection de porteuse (Data carrier detect) RS232 (applicable seulement au module BMX NOM 0200).	%IWr . m . c . 0 . 0
CTS	BOOL	R	Signal prêt à envoyer (Clear to send) RS232.	%IWr . m . c . 0 . 2
DSR	BOOL	R	Signal terminal de données prêt (Data set ready) RS232 (applicable seulement au module BMX NOM 0200).	%IWr . m . c . 0 . 3

NOTE : Quand CTS est vert dans Punit, c'est que %IWr . m . c . 0 . 0 est à 1 et que la tension sur ce signal est positive. La même règle est applicable aux signaux DCD et DSR.

Informations détaillées sur les objets à échange explicite IODDT de type T_COM_CHAR_BMX

Vue d'ensemble

Cette partie présente les objets à échanges explicites de l'IODDT de type T_COM_CHAR_BMX qui s'appliquent à la communication en mode caractère. 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.

Observations

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.

Indicateur 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 (%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-application au module BMX NOM 0200).	%MWr.m.c.0.2

Compte rendu d'échanges explicites : 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	Erreur détectée 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-application au module BMX NOM 0200).	%MWr.m.c.1.2

Défaillances détectées sur la 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 communication nécessaire sur l'automate.	%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 spécifique de la voie, %MWr.m.c.3

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

Symbole standard	Type	Accès	Signification	Adresse
PROTOCOL	INT	L	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 la 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 (Data Terminal Ready).	%MWr.m.c.24.8
DTR_OFF	BOOL	L/E	Réinitialise le signal terminal de données prêt (Data Terminal Ready).	%MWr.m.c.24.9

La commande est exécutée par l'instruction WRITE_CMD (IODDT_VAR1).

Pour plus d'informations sur la modification des protocoles, reportez-vous à la section relative aux changements de protocole ([voir page 262](#)).

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	Arrêter tous les échanges sur le front montant (disponible 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
%KWrr.m.c.0	INT	R	L'octet 0 de ce mot correspond au type. La valeur 3 représente le mode caractère.

Adresse	Type	Accès	Signification
%KWr.m.c.1	INT	R	L'octet 0 de ce mot correspond à la vitesse de transmission. Il peut avoir différentes valeurs : ● La valeur -2 (0xFE) correspond à 300 bits/s. ● La valeur -1 (0xFF) correspond à 600 bits/s. ● La valeur 0 (0x00) correspond à 1200 bits/s. ● La valeur 1 (0x01) correspond à 2400 bits/s. ● La valeur 2 (0x02) correspond à 4800 bits/s. ● La valeur 3 (0x03) correspond à 9600 bits/s (valeur par défaut). ● La valeur 4 (0x04) correspond à 19200 bits/s. ● La valeur 5 (0x05) correspond à 38400 bits/s. ● La valeur 6 (0x06) correspond à 57600 bits/s (ne peut être utilisé que pour le module BMX NOM 0200) ● La valeur 7 (0x07) correspond à 115200 bits/s (ne peut être utilisé que pour le module BMX NOM 0200) L'octet 1 de ce mot correspond au format : ● Bit 8 : nombre de bits (1 = 8 bits (RTU), 0 = 7 bits (ASCII)) ● bit 9 = 1 : gestion de la parité (1 = avec, 0 = sans) ● Bit 10 : type de parité (1 = impair, 0 = pair) ● Bit 11 : nombre de bits d'arrêt (1 = 1 bit, 0 = 2 bits) ● Bit 13 : ligne physique (1 = RS232, 0 = RS485) ● 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) ● 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).
%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).

Adresse	Type	Accès	Signification
%KWr.m.c.3	INT	R	Ce mot correspond au type de polarisation : ● 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) ● 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 ● 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
%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 : ● 1, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 1 est activée ● 0, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 1 est désactivée Deux valeurs peuvent être associées au bit 1 de l'octet 0 : ● 1, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 1 est activée ● 0, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 1 est désactivée 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.

Adresse	Type	Accès	Signification
%KWr.m.c.7	INT	R	Deux valeurs peuvent être associées au bit 0 de l'octet 0 : • 1, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 2 est activée • 0, qui implique que l'option d'arrêt de la zone Arrêt en réception pour le caractère 2 est désactivée Deux valeurs peuvent être associées au bit 1 de l'octet 0 : • 1, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 2 est activée • 0, qui implique que l'option Caractère inclus de la zone Arrêt en réception pour le caractère 2 est désactivée 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 10.5

IODDT de type T_GEN_MOD applicable à tous les modules

Détails des objets langage du IODDT de type T_GEN_MOD

Introduction

Tous les modules des automates Modicon M340 et X80 disposent d'un IODDT associé 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	R	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	R	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	R	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	R	module inutilisable	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Voie(s) inutilisable(s)	%MWr.m.MOD.2.1
BLK	BOOL	R	Bornier incorrectement câblé	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Anomalie de configuration matérielle ou logicielle	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module absent ou inopérant	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Mot d'erreurs internes détectées du module (extension Fipio uniquement)	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Erreur interne détectée, module hors service (extension Fipio uniquement)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Voie(s) inutilisable(s) (extension Fipio uniquement)	%MWr.m.MOD.2.9

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

Sous-chapitre 10.6

Objets langage et DDT d'équipement associés aux communications Modbus

Noms de DDT d'équipements de communication

Introduction

Cette rubrique décrit les **DDT d'équipements de communication** de Unity Pro.

Le nom du DDT par défaut des équipements contient les informations suivantes :

- Entrée et/ou sortie du module (symbole **X**)
- Numéro d'insertion du module (symbole **#**)

Exemple : MOD_COM_X_#

Le type du DDT par défaut des équipements contient les informations suivantes :

- Plateforme avec :
 - M pour Modicon M340 ou Modicon M580
- Type d'équipement (COM pour TOR)
- Fonction (NOM pour BMX NOM 0200)
- Sens :
 - IN
 - OUT

Liste des DDT d'équipements implicites

Le tableau ci-dessous présente la liste des équipements pris en charge par Modicon M340 et par Modicon M580, ainsi que le nom et le type DDT d'équipement correspondant :

Nom du DDT d'équipement	Type du DDT d'équipement	Equipements Modicon M340/M580
MOD_NOM_#	T_M_COM_NOM	BMX NOM 0200.3

Description des DDT d'équipements implicites

Le tableau suivant indique les bits du mot d'état T_M_COM_NOM :

Symbole standard	Type	Signification	Accès
MOD_HEALTH	BOOL	0 = le module a détecté une erreur 1 = le module fonctionne correctement	Lecture
MOD_FLT	BYTE	Octet d'erreurs internes détectées du module	Lecture
COM_CH	ARRAY [0...1] of T_M_COM_NOM_CH	Tableau de structure	

Le tableau suivant indique les bits du mot d'état T_M_COM_NOM_CH[0...1] :

Symbole standard		Type	Bit	Signification	Accès
FCT_TYPE		WORD		0 = la voie n'est pas utilisée	Lecture
				1 = la voie est utilisée	
				3 = mode caractère	
				7 = esclave MODBUS	
CH_HEALTH		BOOL		0 = une erreur est détectée sur la voie	Lecture
				1 = la voie fonctionne correctement	
INPUT_SIGNALS [INT]	DCD	BOOL	0	Signal détection de porteuse (DCD, Data Carrier Detect) RS232 (applicable seulement au module BMX NOM 0200)	Lecture
	CTS	BOOL	2	Prêt à envoyer le signal RS232	Lecture
	DSR	BOOL	3	Signal terminal de données prêt (DSR, Data Set Ready) RS232 (applicable seulement au module BMX NOM 0200)	Lecture
COMMAND [INT]	STOP_EXCH	BOOL	0	Front montant à 1 : tous les échanges en cours sont arrêtés.	Lecture / écriture

Description des instances DDT d'équipements explicites

Les échanges explicites (état de lecture), applicables uniquement aux voies d'E/S Modicon M340 et Modicon M580, sont gérés par l'instance d'EFB READ_STS_QX.

- L'adresse de voie ciblée (ADDR) peut être gérée par l'EF ADDMX (*voir Unity Pro, Communication, Bibliothèque de blocs*) (en connectant ADDMX OUT à ADDR).
- Le paramètre de sortie READ_STS_QX (*voir Unity Pro, Gestion des E/S, Bibliothèque de blocs*) (STATUS) peut être connecté à une instance de DDT T_M_xxx_yyy_CH_STS (variable à créer manuellement), où :
 - xxx correspond au type d'équipement,
 - yyy correspond à la fonction.

Exemple : T_M_COM_NOM_CH_STS

Le tableau suivant indique les bits du mot d'état T_M_COM_NOM_CH_STS :

Type	Type	Accès
STRUCT	T_M_COM_NOM_CH_STS	

Le tableau suivant indique les bits du mot d'état de structure T_M_COM_NOM_CH_STS :

Symbole standard		Type	Bit	Signification	Accès
CH_FLT [INT]	NO_DEVICE	BOOL	0	Aucun équipement ne fonctionne sur la voie	Lecture
	ONE_DEVICE_FLT	BOOL	1	Équipement inopérant sur la voie	Lecture
	BLK	BOOL	2	Défaut du bornier détecté (non connecté)	Lecture
	TO_ERR	BOOL	3	Erreur de timeout détectée (câblage défectueux)	Lecture
	INTERNAL_FLT	BOOL	4	Erreur détectée en interne ou autotest de la voie	Lecture
	CONF_FLT	BOOL	5	Anomalie de configuration détectée : différentes configurations logicielle et matérielle	Lecture
	COM_FLT	BOOL	6	Problème lors de la communication avec l'automate (PLC)	Lecture
	APPLI_FLT	BOOL	7	Erreur d'application détectée (erreur de réglage ou de configuration)	Lecture
PROTOCOL		BYTE		6 pour le maître Modbus, 3 pour le mode caractère	Lecture
ADDRESS		BYTE		Adresse de l'esclave	Lecture

Chapitre 11

Changement dynamique de protocoles

Objet de ce chapitre

Ce chapitre présente le changement dynamique de protocoles Modbus et Mode caractère.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Changement de protocole avec les processeurs BMX P34 1000/2000/2010/20102/2020	260
Changement de protocole avec le module BMX NOM 0200	262

Changement de protocole avec les processeurs BMX P34 1000/2000/2010/20102/2020

Général

Cette partie présente la manière de changer le protocole utilisé par une communication série d'UC à l'aide de la commande `WRITE_CMD( IODDT_VAR1 )`. Cette commande permet un changement de protocole parmi les 3 protocoles suivants :

- Esclave Modbus
- Modbus maître
- Mode caractère

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

Changement de protocole : Principe

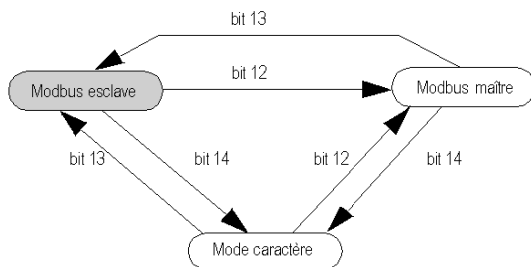
Vous devez d'abord créer une variable `IODDT` associée à la voie série du processeur, puis définir sur 1 le bit du mot `IODDT_VAR1.CONTROL (%MWr . m . c . 24)` qui correspond au changement de protocole souhaité :

- `TO_MODBUS_MASTER` (bit 12) : changement du protocole courant vers Modbus maître,
- `TO_MODBUS_SLAVE` (bit 13) : changement du protocole courant en Modbus esclave
- `TO_CHAR_MODE` (bit 14) : changement du protocole courant en mode caractère

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

Appliquez ensuite l'instruction `WRITE_CMD` à la variable `IODDT` associée à la voie de communication série du processeur.

Le schéma ci-dessous présente les changements de protocole en fonction des bits du mot `IODDT_VAR1.CONTROL (%MWr . m . c . 24)` paramétrés sur 1 :



NOTE : Les changements d'un protocole à l'autre nécessitent que le processeur soit initialement configuré en Modbus esclave.

Cas d'utilisation

Il y a 3 changements de protocole utilisés :

- transfert vers le protocole Modbus maître : le changement de protocole se fait en 2 étapes :
 - transfert de la configuration Modbus esclave vers la configuration Modbus maître,
 - retour en configuration initiale Modbus esclave.

Le but de la configuration Modbus maître est d'envoyer une information à un autre automate à propos d'un événement. Lorsqu'une configuration Modbus esclave est remplacée par une configuration Modbus maître, les réglages de transmission et de configuration, ainsi que les paramètres liés aux signaux et à la ligne physique, demeurent inchangés. Seules les valeurs des paramètres suivants spécifiques à la configuration Modbus maître changent :

- délai entre les trames paramétré sur sa valeur par défaut, qui dépend de la vitesse de transmission
- délai de réponse paramétré sur 3000 ms
- nombre de réitérations réglé sur 3
- Transfert vers le protocole Mode caractère : le changement de protocole se fait en 2 étapes :
 - transfert de la configuration Modbus esclave vers la configuration Mode caractère,
 - retour en configuration initiale Modbus esclave.

Le but de la configuration Mode caractère est de communiquer avec un protocole privé (un modem par exemple). Lorsqu'une configuration Modbus esclave est remplacée par une configuration Mode caractère, les réglages de transmission et de configuration, ainsi que les paramètres liés aux signaux et à la ligne physique, demeurent inchangés. Seul le paramètre de fin de message propre au protocole Mode caractère est réglé sur Arrêt sur silence avec un timeout de 1000 ms.

- Transfert vers les protocoles Mode caractère et Modbus maître : le changement de protocole se fait en 3 étapes :
 - transfert de la configuration Modbus esclave vers la configuration Mode caractère,
 - transfert de la configuration Mode caractère vers la configuration Modbus maître,
 - retour en configuration initiale Modbus esclave.

Le but de la configuration Mode caractère est de communiquer avec un protocole privé (un modem par exemple). Une fois que l'échange est fini, l'utilisateur se met en configuration Modbus maître dans le but d'envoyer une information à un autre automate à propos d'un événement. Après l'envoi de son message, il rétablit le mode de configuration initial Modbus esclave.

NOTE : Dans les 3 cas d'utilisation, le protocole Modbus esclave est celui utilisé par défaut.

Démarrage à froid et démarrage à chaud

Le changement de protocole n'est pas affecté par les bits %S0 (bit à 1 lors d'un démarrage à froid) et %S1 (bit à 1 lors d'un démarrage à chaud). Cependant, un démarrage à froid ou à chaud de l'automate configure le port série avec ses valeurs par défaut ou avec les valeurs programmées dans l'application.

Changement de protocole avec le module BMX NOM 0200

Général

Cette partie présente la manière de changer le protocole utilisé par une communication série BMX NOM 0200 par la commande `WRITE_CMD ( IODDT_VAR1 )`.

Cette commande permet un changement de protocole parmi les 3 protocoles suivants :

- Esclave Modbus
- Modbus maître
- Mode caractère

NOTE : la variable `IODDT_VAR1` doit être de type `T_COM_MB_BMX` ou `T_COM_MB_BMX CONF EXT`.

Changement de protocole : Principe

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

- `TO_MODBUS_MASTER` (bit 12) : changement du protocole courant vers Modbus maître,
- `TO_MODBUS_SLAVE` (bit 13) : changement du protocole courant en Modbus esclave
- `TO_CHAR_MODE` (bit 14) : changement du protocole courant en mode caractère

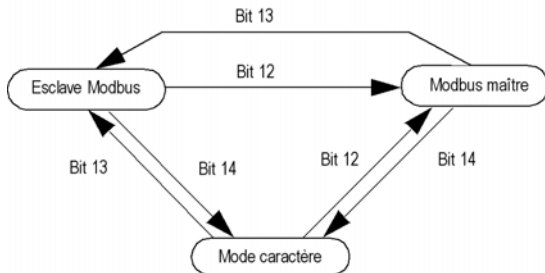
NOTE : Un seul bit peut être mis à 1 à la fois : la mise à 1 de plusieurs bits conduit à une erreur.

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

Appliquez ensuite l'instruction `WRITE_CMD` à la variable `IODDT` associée à la voie série.

NOTE : notez que deux maîtres (sur le même bus) n'envoient pas de requêtes simultanément ; sinon, les requêtes seraient perdues et chaque rapport obtiendrait un résultat erroné qui pourrait être `16#0100` (impossible de traiter la requête) ou `16#ODFF` (esclave non présent).

Le schéma ci-dessous présente les changements de protocole en fonction des bits du mot `IODDT_VAR1.CONTROL (%MWr . m . c . 24)` paramétrés sur 1 :



Cas d'utilisation

Il y a 3 changements de protocole utilisés :

- Transfert d'esclave Modbus vers maître Modbus :
Le but de la configuration Modbus maître est d'envoyer une information à un autre automate à propos d'un événement. Lorsqu'une configuration Modbus esclave est remplacée par une configuration Modbus maître, les réglages de transmission et de configuration, ainsi que les paramètres liés aux signaux et à la ligne physique, demeurent inchangés. Seules les valeurs des paramètres suivants spécifiques à la configuration Modbus maître changent :
 - délai entre les trames paramétré sur sa valeur par défaut, qui dépend de la vitesse de transmission
 - délai de réponse à 3 s
 - nombre de répétitions réglé sur 0
- Transfert d'esclave/maître Modbus vers mode caractère
La commutation vers le mode caractère s'utilise pour envoyer des commandes AT à un modem. Lorsqu'une configuration Modbus est remplacée par une configuration Mode caractère, les réglages de transmission et de configuration, ainsi que les paramètres liés aux signaux et à la ligne physique, demeurent inchangés. Seul le paramètre de détection de fin de message spécifique du mode caractère est mis à l'arrêt à la réception du caractère de terminaison x0d.
- Transfert de mode caractère à maître Modbus et à esclave Modbus :
Le but de la configuration Mode caractère est de communiquer avec un protocole privé (un modem par exemple). Quand l'échange est terminé, l'utilisateur passe à la configuration maître Modbus (avec la temporisation de réponse réglée à 3 s et le nombre de tentatives à 0) de façon à envoyer des informations sur un événement à un autre automate. Après l'envoi de son message, il rétablit le mode de configuration Modbus esclave : le numéro d'esclave est réglé à la valeur enregistrée dans la mémoire FLASH ou à 248 à défaut.

Démarrage à froid et démarrage à chaud

Le changement de protocole n'est pas affecté par les bits %S0 (bit à 1 lors d'un démarrage à froid) et %S1 (bit à 1 lors d'un démarrage à chaud). Cependant, un démarrage à froid ou à chaud de l'automate configure le port série avec ses valeurs par défaut ou avec les valeurs programmées dans l'application.

NOTE : la configuration par défaut du module est la suivante : pour être facilement configurable par un ordinateur tel qu'un PC, la voie 0 est configurée en mode esclave RS232 et la voie 1 en mode RS485. Les autres paramètres sont : 19200 bauds, RTU, parité paire, 1 bit d'arrêt, sans contrôle de flux, temporisation de trame par défaut 1,75 ms, numéro d'esclave 248.

Partie IV

Démarrage rapide : BMX NOM 0200.4

Présentation

Cette section décrit la procédure de configuration du module BMX NOM 0200.4 comme Modbus RS-485 RTU esclave dans une station Modicon X80 sur un automate Quantum.

NOTE : Pour plus de détails sur la manière d'installer et de configurer le BMX NOM 0200.4 dans une station M580 RIO, reportez-vous à *Configuration du module BMX NOM 0200.4 dans une station X80* (voir *Modicon M580, Modules RIO, Guide d'installation et de configuration*).

Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
12	Présentation	267
13	Configuration dans Unity Pro	273

Chapitre 12

Présentation

Conditions requises

Pour configurer le BMX NOM 0200.4, procédez comme suit :

- Utilisez les versions de micrologiciel suivantes :
 - BMX CRA 312 10 : SV $\geq$ V2.10
 - BMX NOM 0200 : SV $\geq$ V1.4
- Reliez un 140 NOC 78• 00 au Quantum 140 CRP 312 00

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation du produit	268
Présentation de l'architecture	269
Limitations	271

Présentation du produit

Présentation

BMX NOM 0200.4 est un nouveau module générique disponible dans la famille de produits liés à la communication du catalogue matériel de Unity Pro et destiné aux seules plates-formes Quantum. Vous devez commencer par sélectionner un module de communication de la station BMX CRA 312 10.3.

Protocoles pris en charge

Pour les modules BMX NOM 0200 :

- RS232 ou RS485 pour la voie 0 ;
- RS485 uniquement pour la voie 1.

La déclaration du module BMX NOM 0200 comme module BMX NOM 0200.4 dans Unity Pro permet de configurer le module de l'une des façons suivantes :

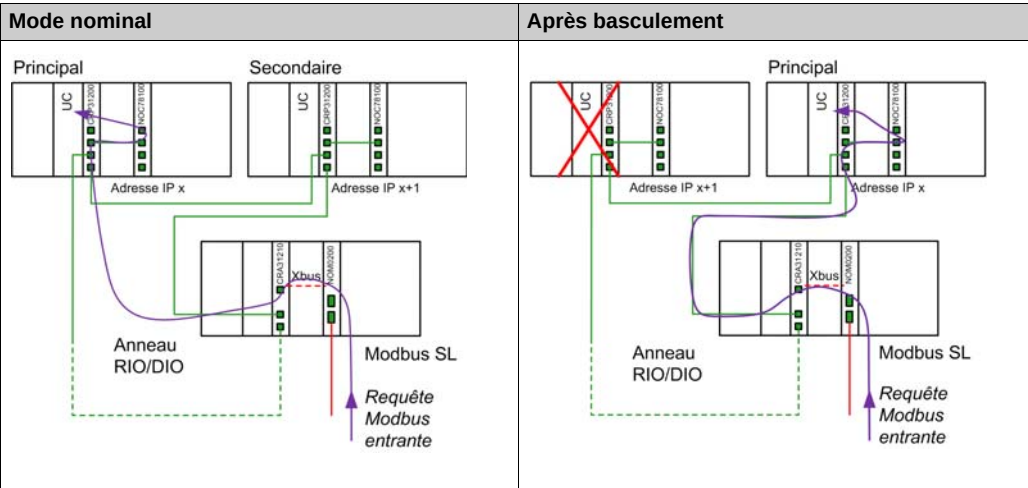
- Modbus RTU esclave sur RS-485
- Modbus série RTU et ASCII maître sur RS-232 et RS-485
- Mode caractère

Compatibilité

Cette offre est compatible avec l'offre standard : BMX NOM 0200, BMX CRA 312 10, 140 CRP 312 00 et UC Quantum.

Cas particulier de la redondance d'UC

L'adresse IP du module 140 NOC 78• 00 bascule en cas de basculement de l'automate. Les requêtes Modbus sont alors acheminées vers l'UC opérationnelle :



NOTE : L'application client Modbus gère l'itération des requêtes en cas de perte de messages pendant le basculement d'un automate.

Limitations

Configuration maximale

Ce tableau indique la configuration maximale du BMX NOM 0200.4 :

Elément	Configuration maximale
Voie maître	4 par station configurée avec un maximum de 36 voies expertes par station. NOTE : Chaque voie configurée du BMX NOM 0200.4 est comptabilisée comme une voie experte.
Station	4 BMX NOM 0200.4 par station.
Système Quantum	16 BMX NOM 0200
Longueur de trame Modbus	256 octets

Adresse IP

Vous devez configurer l'adresse IP du chemin de routage Modbus de chaque module BMX CRA 312 10.3 qui prend en charge un module BMX NOM 0200.4 esclave Modbus.

Unity Pro n'offre aucun contrôle sur la cohérence de ces adresses IP.

AVERTISSEMENT

FONCTIONNEMENT INATTENDU DE L'EQUIPEMENT

Vérifiez que l'adresse IP est bien celle du Quantum qui prend en charge le serveur Modbus.

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

Protocoles pris en charge

Modbus RTU est le seul protocole esclave pris en charge.

Seul le protocole RS-485 est pris en charge lors de la sélection d'un Modbus esclave.

Codes de fonction Modbus pris en charge

Ce tableau répertorie les codes de fonction Modbus pris en charge par le serveur Quantum :

Liaison à -> Code de fonction Modbus :	Type de variable	Code	Fonction
01	%M	0X	Lecture de l'état des bits de sortie
02	%M	1X	Lecture de l'état des bits des entrées
03	%MW	4X	Lecture des registres de maintien
05	%M	0X	Forçage d'un bit de sortie
04	%MW	3X	Lecture du registre d'entrée
06	%MW	4X	Ecriture d'un registre unique
15	%M	0X	Ecriture de plusieurs bits de sortie
16	%MW	4X	Ecriture dans plusieurs registres
23	%MW	4X	Lecture/écriture de plusieurs registres

Chapitre 13

Configuration dans Unity Pro

Introduction

Les modes de fonctionnement sont pour la plupart identiques à celui des versions précédemment prises en charge de BMX NOM 0200.

Ce chapitre se concentre sur les aspects propres à la configuration du module BMX NOM 0200.4 dans Unity Pro.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Insertion de modules	274
Ecran de configuration du module	275

Insertion de modules

Présentation

Dans l'architecture d'E/S Ethernet Quantum, seuls des modules BMX NOM 0200.4 peuvent être insérés dans une station distante Modicon X80, avec le module adaptateur BMX CRA 312 10.3.

Procédure

Procédez comme suit pour insérer le module BMX NOM 0200.4 sur une station distante Modicon X80 :

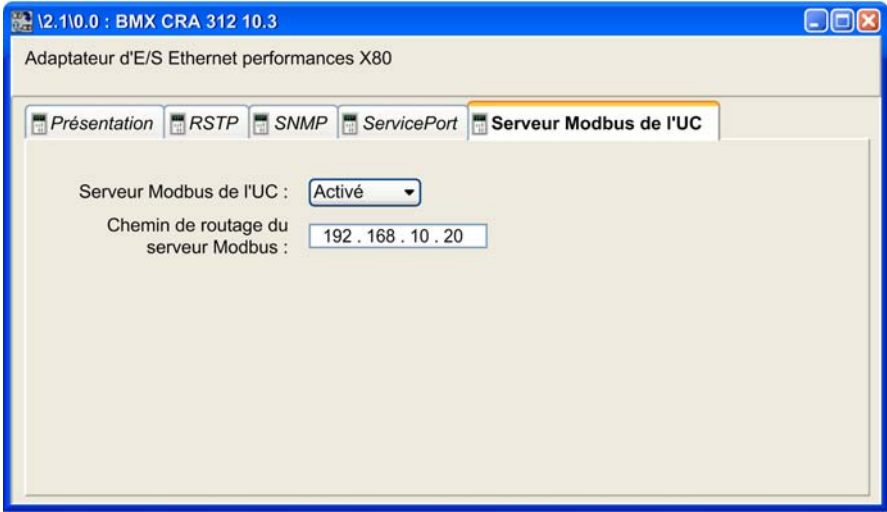
Etape	Action
1	Insérez le module 140 CRP 312 00 dans un rack Quantum local.
2	Sur le Bus EIO , créez une station d'E/S Ethernet Modicon X80 avec un module BMX CRA 312 10.3.
3	Insérez le nouveau module BMX NOM 0200.4 dans la station.
4	Insérez le module 140 NOC 78• 00 dans un rack Quantum local.

Ecran de configuration du module

Configuration du chemin de routage du serveur Modbus

Cette configuration ne peut s'effectuer qu'en mode local (automate non connecté).

Pour définir le chemin de routage du serveur Modbus, procédez comme suit :

Etape	Action
1	Double-cliquez sur le module BMX CRA 312 10.3 dans l'éditeur du configurateur.
2	Sélectionnez l'onglet Serveur Modbus de l'UC . 
3	Sélectionnez la valeur Activé pour le champ Serveur Modbus de l'UC .
4	Entrez l'adresse IP du module 140 NOC 78• 00 dans le champ Modbus server routing path (Chemin de routage du serveur Modbus). Le module 140 NOC 78• 00 gère le routage des trames Modbus entre les E/S Ethernet et l'UC.

Unity Pro n'offre aucun contrôle sur la cohérence de ces adresses IP.

AVERTISSEMENT

FONCTIONNEMENT INATTENDU DE L'EQUIPEMENT

Vérifiez que l'adresse IP est bien celle du Quantum qui prend en charge le serveur Modbus.

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

NOTE : Vous devez configurer l'adresse IP du chemin de routage Modbus de chaque module BMX CRA 312 10.3 qui prend en charge un module BMX NOM 0200.4 esclave Modbus.

Accès aux écrans de configuration des voies

Pour accéder aux écrans de configuration des voies du module BMX NOM 0200.4, procédez comme suit :

Etape	Action
1	Ouvrez le sous-répertoire BMX NOM 0200.4 dans le navigateur du projet.
2	Sélectionnez la voie à configurer. Par défaut : ● La Voie 0 est configurée avec la fonction Character mode link (liaison en mode caractère). ● La Voie 1 est configurée avec la fonction Modbus link (liaison Modbus). NOTE : Certains paramètres dont l'affichage est estompé ne sont pas accessibles.

Pour configurer la communication Modbus Serial en mode maître, reportez-vous au chapitre *Communications Modbus Serial du BMX NOM 0200* ([voir page 131](#)).

Pour configurer la communication en Mode caractère, reportez-vous au chapitre *Communication en Mode caractère du BMX NOM 0200* ([voir page 179](#)).

Ecran de configuration de liaison Modbus esclave

La figure ci-dessous illustre l'écran de configuration esclave du module BMX NOM 0200.4 :

The screenshot displays the configuration window for the BMX NOM 0200.4 module, titled "12.110.2 : BMX NOM 0200.4". The window is divided into several sections:

- Left Panel:** Shows a tree view with "BMX NOM 0200.4 (MOD_COM_2_1)" expanded, revealing "Voie 0" and "Voie 1".
- Top Section:** "Bus Module 2 RS485/232 port (SV >=V1.4)".
- Configuration Tab:**
 - Type:** A dropdown menu set to "Esclave".
 - Maître:** Includes "Nombre de répétitions" (0) and "Délai de réponse" (1 X 10 ms).
 - Esclave:** Includes "Numéro d'esclave" (1) and an "Externe" checkbox.
 - Ligne physique:** Radio buttons for "RS232" and "RS485" (selected).
 - Signaux:** Radio buttons for "RX/TX", "RX/TX • RTS/CTS", "DTE mode", "RX/TX • RTS/CTS", "DCE mode", "RX/TX • RTS/CTS", and "•DTR/DSR/DCO".
 - Vitesse de transmission:** A dropdown menu set to "19200 bits/s".
 - Délai inter-trames:** A checkbox for "Par défaut" (checked) and a value of "2 ms".
 - Données:** Radio buttons for "ASCII (7 bits)", "RTU (8 bits)" (selected), and "Arrêt".
 - Arrêt:** Radio buttons for "1 bit" (selected) and "2 bits".
 - Parité:** Radio buttons for "Paire" (selected), "Impaire", and "Aucun".
 - Délai RTS/CTS:** A value of "0 X 100 ms".
- Bottom Section:** Includes "Fonction : Liaison Modbus" and "Tâche : MAST".

Le tableau ci-dessous contient les valeurs par défaut des paramètres de l'écran de configuration Modbus esclave :

Paramètre de configuration	Valeur par défaut
Type	Esclave
Numéro d'esclave	1
Ligne physique	RS-485 uniquement
Signaux	RX/TX uniquement
Vitesse de transmission	19200 bits/s
Délai entre les trames	2 ms
Données	8 bits uniquement
Arrêter	1 bit
Parité	Paire

NOTE : Modbus est un protocole standard. Ce module repose sur un mode d'échange de données unique.

Lorsque vous configurez la communication Modbus Serial en mode maître, les paramètres esclave sont grisés et non modifiables.

Partie V

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
14	Description de l'application	281
15	Installation de l'application avec Unity Pro	283
16	Démarrage de l'application	311

Chapitre 14

Description de l'application

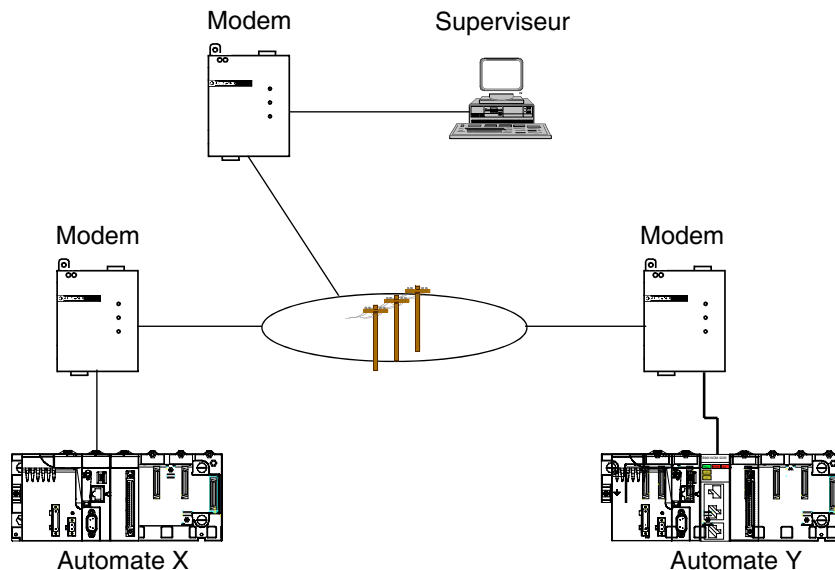
Vue d'ensemble de l'application

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 15

Installation de l'application avec Unity Pro

Objet de ce 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
15.1	Présentation de la solution utilisée	284
15.2	Développement de l'application	285

Sous-chapitre 15.1

Présentation de la solution utilisée

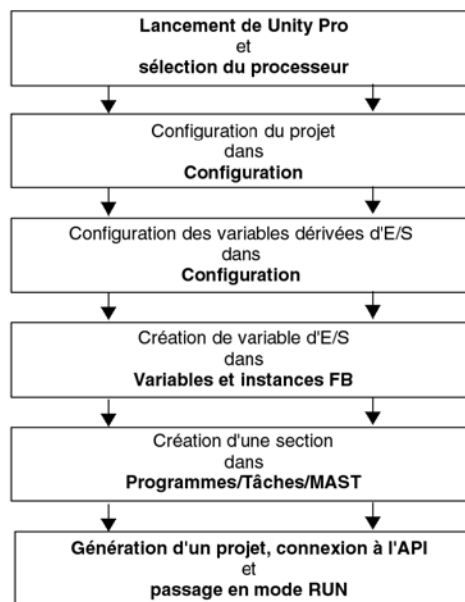
Différentes étapes du processus utilisant Unity Pro

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 15.2

Développement de l'application

Objet de cette section

Ce sous-chapitre décrit pas à pas la création de l'application à l'aide de Unity Pro.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Création du projet	286
Déclaration des variables	291
Utilisation d'un modem	295
Procédure de programmation	297
Structure de programmation	299
Programmation	302

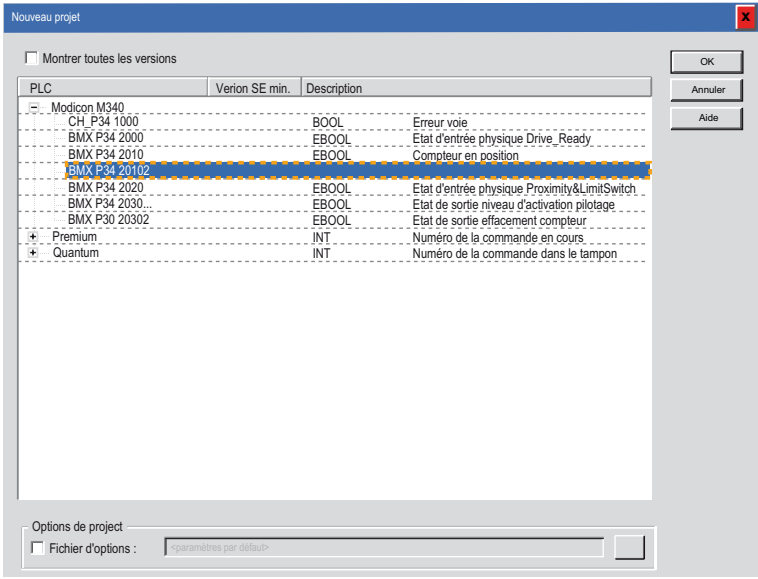
Création du projet

Présentation

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

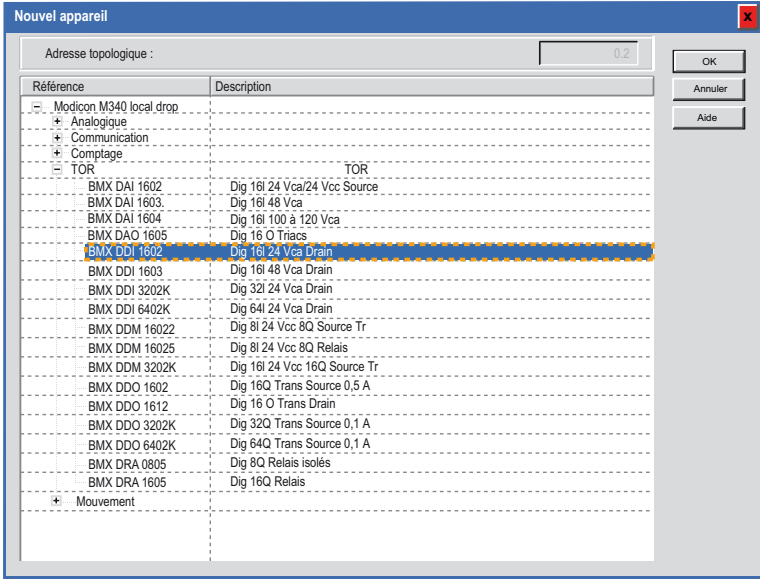
Marche à suivre pour créer un projet

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

Etape	Action
1	Lancez le logiciel Unity Pro.
2	Cliquez sur Fichier, puis sur Nouveau et choisissez un processeur BMX P34 20102 : 
3	Cliquez sur OK pour valider.

Sélection de module d'entrée TOR

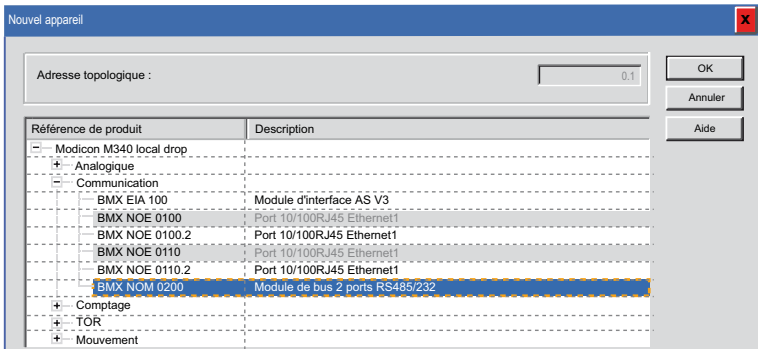
Le tableau ci-dessous présente la procédure de sélection du module TOR dont a besoin l'automate X.

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 correspond au numéro de rack).
2	Dans la fenêtre Bus d'automate, sélectionnez un emplacement (par exemple, emplacement 1) et cliquez deux fois dessus.
3	Choisissez le module d'entrée TOR BMX DDI 1602 situé dans la liste de modules TOR, 
4	Cliquez sur OK pour valider.

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. En conséquence il doit être ajouté 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 correspond au numéro de rack).
2	Dans la fenêtre Bus d'automate, sélectionnez un emplacement (par exemple, emplacement 1) et cliquez deux fois dessus.
3	Choisissez le module de communication BMX NOM 0200 situé dans la liste de modules Communication, 
4	Cliquez sur OK pour valider.

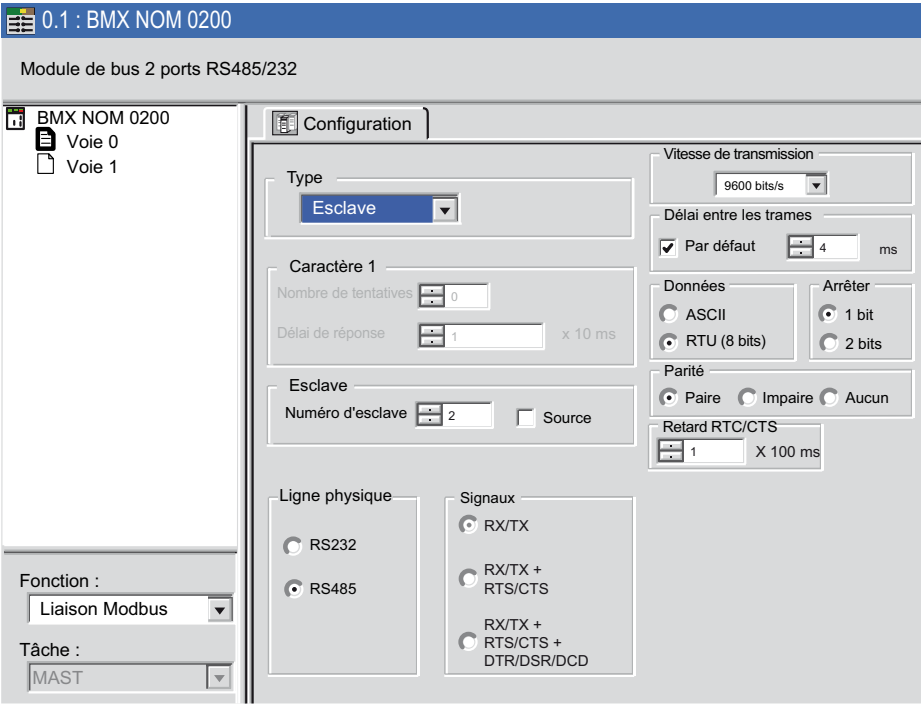
Configuration du port série du processeur

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	Dans le Navigateur du projet, cliquez deux fois sur Configuration, puis sur 0 : BMX XBP 0800 et enfin sur 0 : BMX P34 20102. Cliquez deux fois sur Port série pour accéder à la fenêtre Port série 0.0:0.
2	Sélectionnez le type Esclave.
3	Sélectionnez 9600 bits/s pour la vitesse de transmission.
4	Sélectionnez RS232 pour la ligne physique.
5	Sélectionnez RTU (8 bits) pour le type de données.
6	Fermez la fenêtre et validez en cliquant sur OK.

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 d'automate X en tant qu'esclave Modbus :

Etape	Action
1	Dans le Navigateur de projet, faites un double-clic sur Configuration puis sur 0 : BMX XBP 0800 puis sur 0 : BMX NOM 0200 pour accéder à la fenêtre 0.x : BMX NOM 0200 (où x est le numéro de logement, par exemple x=1). 
2	Sélectionnez la Voie 0.
3	Sélectionnez la Liaison Modbus pour la fonction.
4	Sélectionnez le type Esclave.
5	Sélectionnez 9600 bits/s pour la vitesse de transmission.
6	Sélectionnez RS232 pour la ligne physique.
7	Sélectionnez RX/TX + RTS/CTS + DTR/DSR/DCD pour les signaux.
8	Sélectionnez 100 ms pour la temporisation RTS/CTS.
9	Sélectionnez RTU (8 bits) pour le type de données.
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, reportez-vous à l'aide en ligne Unity Pro (cliquez sur ?, Unity, Unity Pro Software, Modes opératoires, puis Editeur de données).

Procédure de déclaration des variables

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

Etape	Action
1	Dans Navigateur de projet / Variables et instances FB, double-cliquez sur Variables élémentaires.
2	Dans la fenêtre Editeur de données, cochez la case de la colonne Nom et saisissez 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.

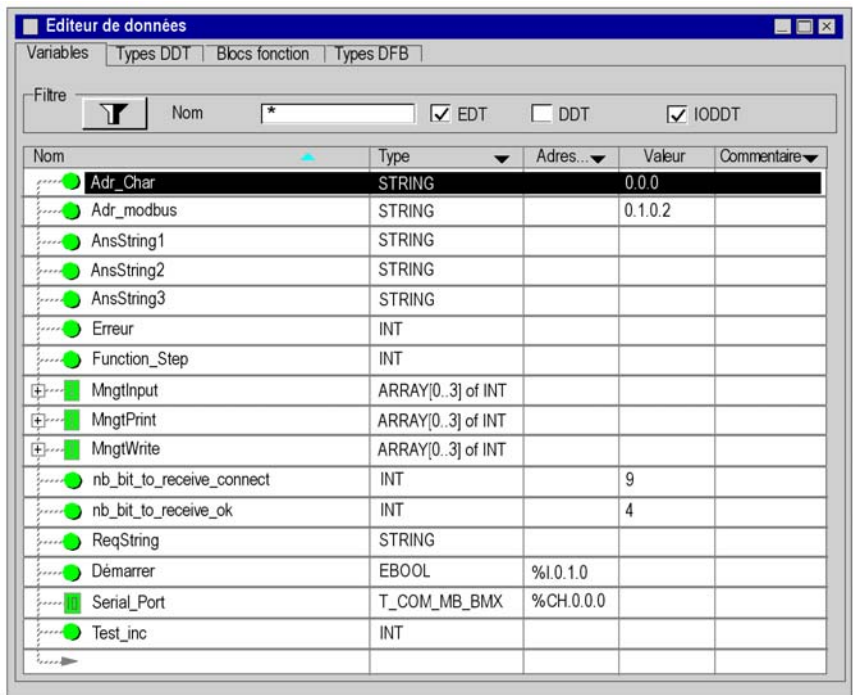
Variables utilisées pour l'application

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

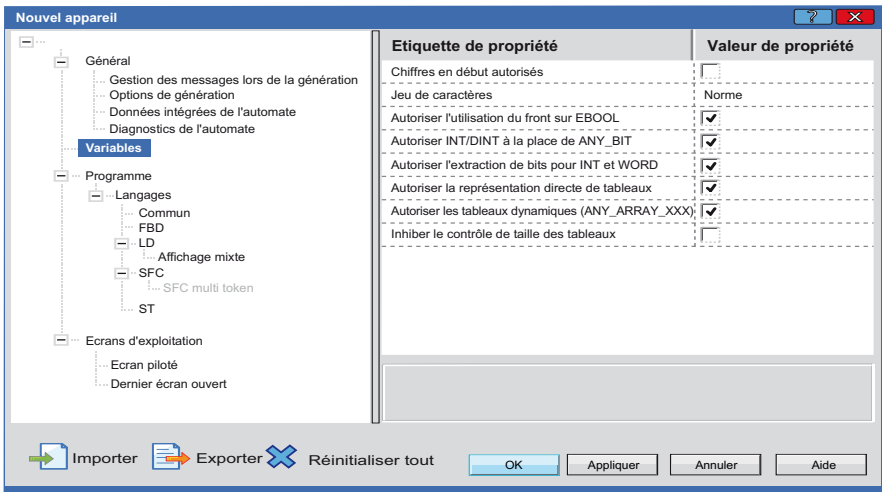
Variable	Type	Définition
nb_charac_to_receive_connect	INT	Nombre de caractères à recevoir : connexion du modem
nb_charac_to_receive_ok	INT	Nombre de caractères à recevoir : message de confirmation du modem
ReqString	STRING	Réponse du modem
Démarrer	EBOOL	Mode de démarrage (signal provenant de la voie 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 :



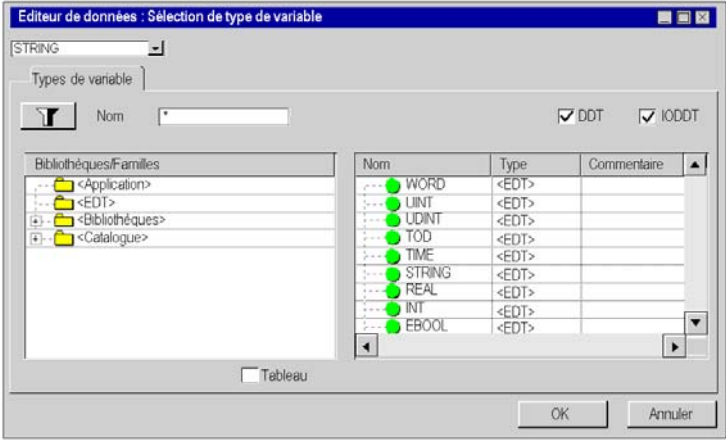
Déclaration d'un type de tableau

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 du projet , cliquez sur Variables et instances FB.
2	Cliquez dans la colonne Nom et entrez un nom pour la variable.

Etape	Action
3	Cliquez deux fois dans la colonne Type, puis cliquez sur le bouton . La fenêtre Sélection de type de variable s'ouvre: 
4	Choisissez le type de variable voulue (cliquez par exemple sur <EDT> et sélectionnez INT), puis cliquez sur la case Tableau. 
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 <Catalog> 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.	● Définissez le Modbus esclave en mode caractère en envoyant WRITE_CMD (<i>voir page 298</i>) au port série. ● Pour une transmission par modem, envoyez la commande HAYES en utilisant le bloc PRINT_CHAR pour configurer le modem (<i>voir page 295</i>). ● 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 295</i>).
2	Mode Modbus maître	● Passez en mode Modbus maître à l'aide de la fonction WRITE_CMD (<i>voir page 298</i>). ● Envoyez les données à écrire sur l'automate esclave.
3	Réinitialisation du port de communication.	● Passez en mode caractère à l'aide de la fonction WRITE_CMD (<i>voir page 298</i>). ● 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 296</i>) en utilisant le bloc PRINT_CHAR. ● Revenez au mode initial du port série (Modbus esclave) à l'aide de la commande WRITE_CMD (<i>voir page 298</i>).

Écriture des mots de commande

Les étapes suivantes permettent d'envoyer une commande WRITE_CMD vers un port de communication :

Etape	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. Remarque : 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. Passer 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. ● S'il n'y a pas d'erreur sur le port série ○ et si le mode caractère est actif, passer à l'étape 15. ○ 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. ● S'il y a une erreur sur le port série, alors ○ affecter Erreur à 10. ○ Passer à l'étape 130.	
15	Envoyer une commande de numérotation au modem par le bloc PRINT_CHAR. Passer 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	Élément
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. Passer 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. ● S'il n'y a pas d'erreur sur le port série ○ et si le mode caractère est actif, passer à l'étape 50. ○ 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. ● S'il y a une erreur sur le port série, alors ○ affecter Erreur à 45. ○ Passer à l'étape 130.	
50	Initialisation du paramètre du bloc WRITE_VAR. Envoyer les données à écrire sur l'automate à l'aide de la fonction WRITE_VAR. Passer 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. Passer à l'étape 65	Mode caractères
65	Lire l'état du port série et vérifier qu'aucune commande n'est active. Réinitialiser le bit de commande TO_CHAR_MODE. ● S'il n'y a pas d'erreur sur le port série ○ et si le mode caractère est actif, passer à l'étape 70. ○ 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. ● S'il y a une erreur sur le port série, alors ○ affecter Erreur à 65. ○ Passer à l'étape 130.	

Numéro de l'étape	Description de l'étape	Elément
70	Envoyer une séquence d'échappement au modem à l'aide du bloc PRINT_CHAR. Passer 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. Passer 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. Passer 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. ● S'il n'y a pas d'erreur sur le port série ○ et si le mode caractère est actif, passer à l'étape 130. ○ 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. ● S'il y a une erreur sur le port série, alors ○ affecter Erreur à 125. ○ Passer à l'étape 130.	
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 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(M(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;
                END_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;
    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 16

Démarrage de l'application

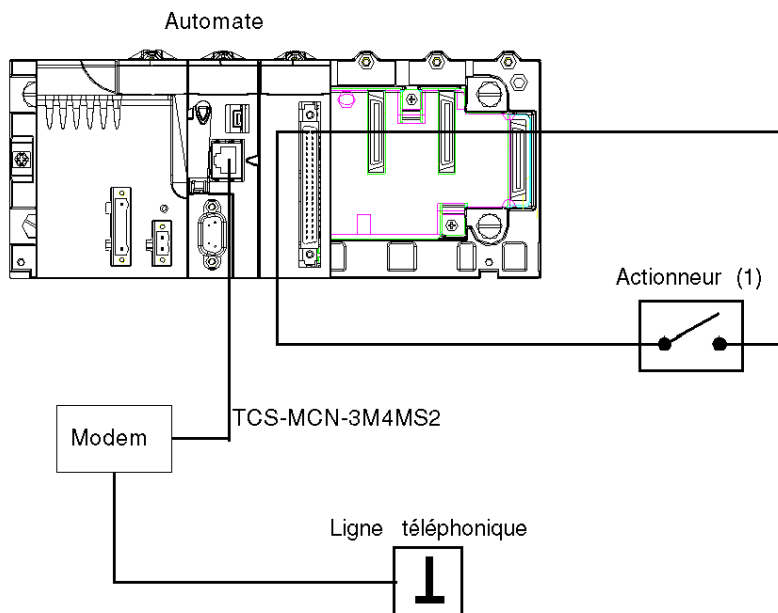
Exécution de l'application en mode Standard

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 :



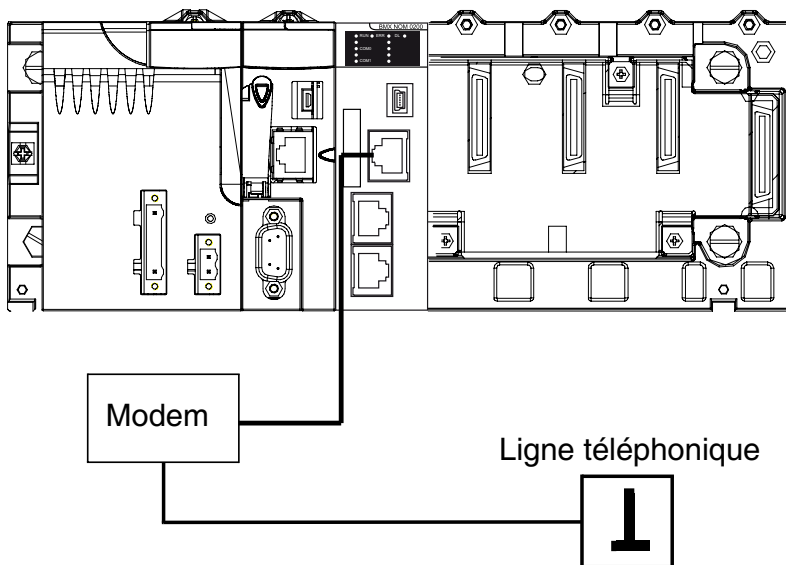
(1) : l'actionneur est connecté à la voie 0 du module TOR.

In this example, the first modem is connected to the processor serial port of the first slave PLC.
The actuator state controls the Start variable state in the application.

Câblage du second automate esclave

Le deuxième automate esclave est raccordé comme suit :

Automate esclave



In this example, the second modem is connected to the channel 0 of the BMX NOM 0200 module of the second slave PLC.

For a better communication reliability, the cable TCS XCN 3M4F3S4 is used for DTR/DSR/DCD modem signals handling by the application.

Configuration of the Second Slave PLC

Before transferring the project for configuring the second slave PLC, verify that the second slave PLC is not connected to the modem.

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

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

Application Transfer to the First Slave PLC

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 Automate, cliquez sur Mode Standard.
2	Dans le menu Génération, cliquez sur Régénérer tout le projet. Votre projet est généré et prêt à être transféré vers l'automate. Lors de la génération du projet, une fenêtre de résultats s'ouvre. Dans le cas où une erreur survient dans le programme, Unity Pro indique son emplacement (lorsque vous cliquez sur la séquence mise en évidence).
3	Dans le menu Automate, cliquez sur Connecter. Vous êtes connecté à l'automate.
4	Dans le menu Automate, cliquez sur Transfert du projet vers l'automate. La fenêtre Transfert du projet vers l'automate s'ouvre. Cliquez sur Transférer. L'application est transférée vers l'automate.

Application Execution on the First Slave PLC

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

Etape	Action
1	Dans la fenêtre Automate, cliquez sur Exécuter. La fenêtre Exécuter s'ouvre. Cliquez sur OK. L'application est maintenant en cours d'exécution sur l'automate.
2	Débranchez du premier automate esclave le PC qui exécute le logiciel Unity Pro.
3	Raccordez le premier automate esclave à un modem SR2MOD01.



!

%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 constante.

%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 un tableau d'éléments de même type. La syntaxe est la suivante : ARRAY [<limites>] OF <Type> Exemple : ARRAY [1..2] OF BOOL est un tableau à une dimension composé de deux éléments de type BOOL. ARRAY [1..10, 1..20] OF INT est un tableau à deux dimensions composé de 10x20 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 7 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 possède l'une ou l'autre des valeurs suivantes : 0 (FALSE) ou 1 (TRUE). Un bit extrait d'un mot est de type BOOL, par exemple : %MW10.4.

BYTE

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

C

Concentrateur

Appareil concentrateur reliant plusieurs modules souples et centralisés 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 de constante de l'automate %KW. L'application de l'automate ne peut pas les modifier.

CPU ou UC

CPU est l'abréviation de Central Processing Unit : nom générique utilisé pour les processeurs Schneider Electric.

CRC

CRC est l'abréviation de Cyclic Redundancy Checksum (contrôle de redondance cyclique) : il signale si un caractère a été "déformé" pendant la transmission de la trame.

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 INTeget (entier double) (codé sur 32 bits). Les limites inférieure et supérieure sont les suivantes : -(2 puissance 31) à (2 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 réalise 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 fonctionnel (instance)]". Contrairement aux appels de bloc fonction, les appels de fonction comportent uniquement une sortie qui n'est pas nommée et dont le nom est identique à celui de la fonction. Dans FBD, chaque appel est indiqué par un [numéro] unique via le bloc graphique. Ce numéro est généré 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(e) 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 INTeRger" (entier simple) (codé sur 16 bits). Les limites inférieure et supérieure sont les suivantes : $-(2 \text{ puissance } 15)$ à $(2 \text{ puissance } 15) - 1$. 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. C'est un témoin qui s'allume quand l'électricité le traverse. Signale l'état de fonctionnement d'un module de communication.

LRC

LRC est l'abréviation de Longitudinal redundancy check ou contrôle de redondance longitudinal : il a été conçu pour détecter la faible probabilité de détection d'erreur du contrôle de parité.

M

Mémoire FLASH

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

Module TOR

Module avec entrées/sorties tout ou rien.

Momentum

Modules d'E/S utilisant plusieurs réseaux de communication à norme ouverte.

P

PLC ou automate

PLC est l'abréviation de Programmable logic controller ou automate programmable. Cerveau d'un processus de fabrication industriel. On dit qu'un tel dispositif "automatise un processus", par opposition à un système de contrôle à relais. Ces automates sont de vrais ordinateurs conçus pour survivre dans les 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 à l'aide de 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

Standard de communication de 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.

Il reste néanmoins, dans le cas d'atténuation éventuelle du signal, la détection assurée jusqu'aux limites -3 V et +3 V. Entre ces deux limites, le signal est considéré comme non valable. Les branchements RS232 sont assez sensibles aux interférences. Le standard précise de ne pas dépasser une distance de 15 m ou 9 600 bauds (bits/s).

RS485

Standard de connexion série qui fonctionne dans un différentiel de 10 V/+5 V. Il utilise deux fils 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

Une méthode de transmission de données capable de communication dans l'un ou l'autre sens, mais 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 de caractères 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 rapide. 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 (point d'accès d'émission) : l'ensemble de connexion au bus.

Trame

Une trame est un groupe de bits constituant un bloc TOR 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

Unity Pro

Logiciel de programmation d'automate Schneider Automation.

V

Variable

Entité mémoire du 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 est utilisé pour effectuer des traitements sur des chaînes de bits.

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

Embase	Limite inférieure	Limite supérieure
Hexadécimal	16#0	16#FFFF
Octal	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 surveillent 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, 61

B

BMXNOM0200, 19
BMXNOM0200.4, 265
BMXP341000, 19
BMXP342000, 19
BMXP342010, 19
BMXP3420102, 19
BMXP342020, 19
bus Modbus, 71

C

Câblage, 61
changement de protocole, 260, 262
configuration des paramètres, 221
configuration Modbus, 79
configuration, mode caractère, 110

E

équipements de connexion, 39

I

INPUT_CHAR, 122, 198

M

M340
 protégé, 32
 renforcé, 32
mise à la terre, 33
mise au point Modbus, 102, 175
mise au point, mode caractère, 128
mise en route, 279
mode caractère, 105

P

PRINT_CHAR, 122, 198
programmation, bus Modbus, 91
programmation, mode caractère, 122

S

structure des données de voie pour commu-
nications Modbus
 T_COM_MB_BMX, 236, 237
structure des données de voie pour le mode
caractère
 T_COM_CHAR_BMX, 245, 246
structure des données de voie pour les proto-
coles de communication
 T_COM_STS_GEN, 231
Structure des données de voie pour les pro-
tocoles de communication
 T_COM_STS_GEN, 232
structure des données de voie pour tous les
modules
 T_GEN_MOD, 253, 253

T

T_COM_CHAR_BMX, 245, 246
T_COM_MB_BMX, 236, 237
T_COM_STS_GEN, 231, 232
T_GEN_MOD, 253, 253
T_M_COM_NOM, 255

