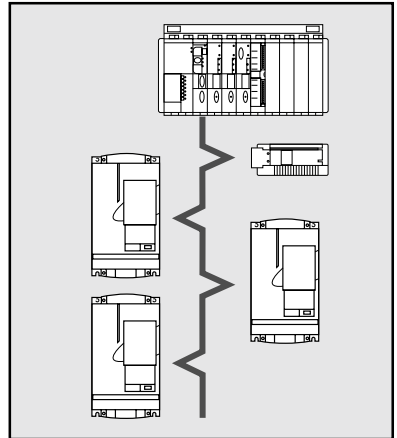


# Altivar 58 Telemecanique

Guide d'exploitation  
User's Manual

Carte de communication  
protocoles UNI-TELWAY,  
Modbus / Jbus  
Communication card  
protocoles UNI-TELWAY,  
Modbus / Jbus

**VW3-A58303**



**GROUPE SCHNEIDER**

---

Carte de communication protocoles UNI-TELWAY, Modbus / Jbus Page 2

---

Communication card protocols UNI-TELWAY, Modbus / Jbus Page 26

---

F  
R  
A  
N  
Ç  
A  
I  
S  
  
E  
N  
G  
L  
I  
S  
H



Lorsque le variateur est sous tension, les éléments de puissance ainsi qu'un certain nombre de composants de contrôle sont reliés au réseau d'alimentation. *Il est extrêmement dangereux de les toucher. Le capot du variateur doit rester fermé.*

Après mise hors tension réseau de l'ALTIVAR, *attendre 3 minutes avant d'intervenir dans l'appareil.* Ce délai correspond au temps de décharge des condensateurs.

Malgré tout le soin apporté à l'élaboration de ce document, Schneider Electric SA ne donne aucune garantie sur les informations qu'il contient, et ne peut être tenu responsable ni des erreurs qu'il pourrait comporter, ni des dommages qui pourraient résulter de son utilisation ou de son application.

Les produits et les additifs présentés dans ce document sont à tout moment susceptibles d'évolutions quant à leurs caractéristiques de présentation et de fonctionnement. Leur description ne peut en aucun cas revêtir un aspect contractuel.

# Sommaire

---

<b>Mise en œuvre matérielle</b>	<b>4</b>
Présentation	4
Installation de la carte, adresse	5
<b>Connexion sur bus multipoint</b>	<b>7</b>
<b>Mise en œuvre logicielle</b>	<b>9</b>
Configuration des fonctions de communication	9
<b>Principe de la communication</b>	<b>10</b>
<b>Requêtes UNI-TELWAY</b>	<b>11</b>
Trame UNI-TELWAY	12
Requête spécifique de commande	13
<b>Protocole Modbus / Jbus</b>	<b>14</b>
Trames Modbus	14
Détails des trames (mode RTU)	20
Mode ASCII	23
<b>Diagnostic</b>	<b>24</b>

# Mise en œuvre matérielle

---

## Présentation

La carte de communication VW3-A58303 permet de raccorder un variateur Altivar® 58 à des réseaux UNI-TELWAY et Modbus / Jbus.

Elle est livrée avec un câble de raccordement, longueur 3 mètres, équipé de deux connecteurs :

- un SUB-D 9 points pour raccordement sur la carte,
- un SUB-D 15 points pour raccordement sur le bus.

Le variateur Altivar 58 peut recevoir des messages de données et y répondre. Cet échange de données permet d'accéder à certaines fonctions de l'Altivar 58 comme :

- Le téléchargement de paramètres de configuration et de réglages,
- La commande et le contrôle,
- La surveillance,
- Le diagnostic.

# Mise en œuvre matérielle

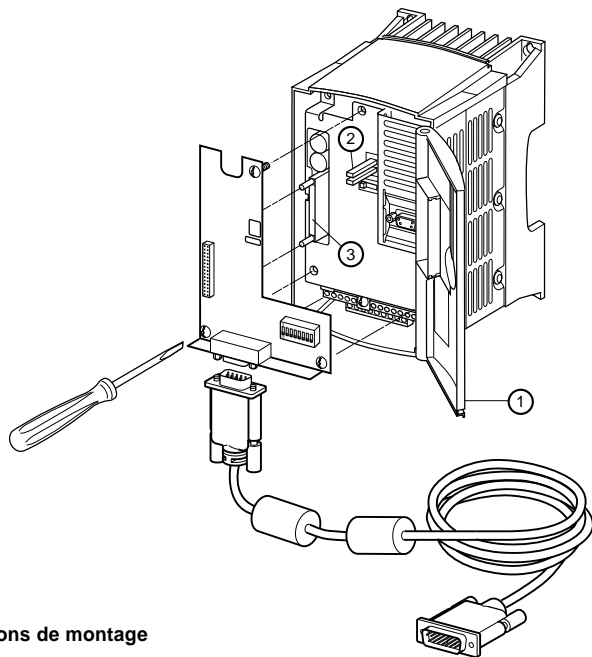
## Installation de la carte, adresse

### Réception

S'assurer que la référence de la carte inscrite sur l'étiquette est conforme au bordereau de livraison correspondant au bon de commande.

Ouvrir l'emballage, et vérifier que la carte option n'a pas été endommagée pendant le transport.

### Installation de la carte dans le variateur



### Précautions de montage

S'assurer que le variateur est hors tension.

Pour accéder à l'emplacement de montage de la carte option, déverrouiller le capot (1) et le faire pivoter.

Contrôler l'absence de tension sur le bus continu : DEL verte (2) (POWER) éteinte, attendre 3 minutes après mise hors tension.

Ecarter le cache de protection flexible du support de la carte contrôle.

Monter la carte option sur le support de la carte contrôle par enfichage sur le connecteur (3), la fixer par ses trois vis.

Le câble de raccordement est à brancher sur la carte par son connecteur 9 points.

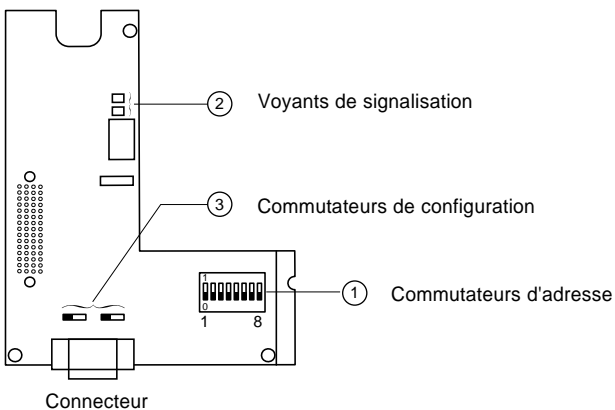
### Mise en place de l'étiquette de visualisation



La carte est fournie avec une étiquette autocollante indiquant la fonction des voyants de signalisation. Coller cette étiquette sur le capot de l'Altivar 58 au dessus de l'étiquette déjà en place, comme représenté ci contre.

# Mise en œuvre matérielle

## Disposition de la carte :



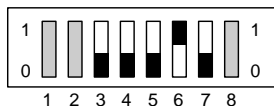
## Codage de l'adresse du variateur :

Un Altivar 58 est identifié sur le bus par son adresse, codée de 0 à 31.

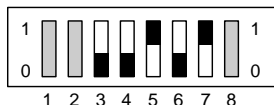
L'adresse correspond au nombre binaire donné par la position 0 ou 1 des 8 commutateurs <sup>1</sup> de la carte (en fait seuls les commutateurs 3 à 7 sont utilisés pour l'adresse).

**Attention :** les bits de poids faibles sont à droite.

Exemples :



adresse 2



adresse 5

Tous les commutateurs à 0 donnent l'adresse 0.

## Voyant lumineux de signalisation :

La carte UNI-TELWAY, Modbus / Jbus est munie de deux voyants de signalisation <sup>2</sup> dont le fonctionnement est détaillé au chapitre diagnostic.

## Commutateurs de configuration :

La carte est munie de 2 commutateurs <sup>3</sup> de configuration du protocole.



• Protocole UNI-TELWAY : (ligne RS485 polarisée)

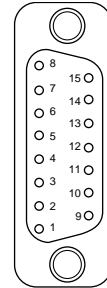
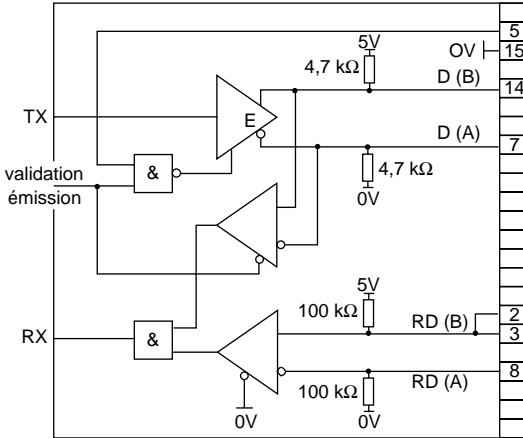


• Protocole Modbus / Jbus : (ligne RS485 non polarisée)

# Connexion sur bus multipoint

## Brochage du connecteur SUB-D

L'interface de transmission conforme au standard RS 485 est isolée galvaniquement du variateur. Elle est disponible sur connecteur SUB-D 15 points.

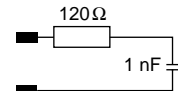
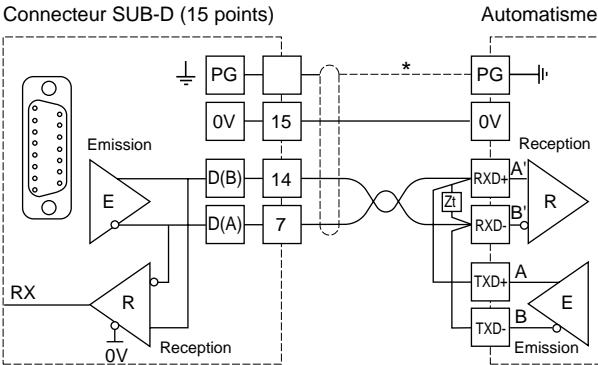


Vue côté contacts extérieurs

## Raccordement sur bus au standard RS 485

### Points à utiliser

Connecteur SUB-D (15 points)



Terminaison de ligne Zt recommandée aux 2 extrémités

\* Le raccordement du blindage aux 2 extrémités dépend des contraintes liées à l'installation

## Recommandations

- utiliser un câble blindé avec 2 paires de conducteurs torsadés,
- relier les potentiels de référence entre eux,
- longueur maximale de la ligne : 1000 mètres,
- longueur maximale d'une dérivation : 20 mètres,
- ne pas connecter plus de 28 stations sur un bus,
- cheminement du câble : éloigner le bus des câbles de puissance (30 cm au minimum), effectuer les croisements à angle droit si nécessaire, raccorder le blindage du câble à la masse de chaque équipement,
- adapter la ligne à ses deux extrémités.





## Configuration des fonctions de communication

### Première mise sous tension

Lors de la première mise sous tension la carte est reconnue automatiquement par l'Altivar 58. Elle donne accès au menu de configuration 8-COMMUNICATION du terminal d'exploitation (ou du terminal de programmation ou du logiciel PC).

### Configuration

Sélectionner le menu 8- COMMUNICATION donnant accès aux paramètres de configuration de la carte.

Ce menu permet de configurer tous les paramètres de communication.

Le premier paramètre est l'adresse du variateur dans le réseau; ce paramètre est accessible uniquement en lecture sur le terminal, car il est configurable matériellement sur la carte par les commutateurs 3 à 7. Le second paramètre est le protocole choisi.

La modification des paramètres de communication n'est possible que quand le moteur est à l'arrêt.

Paramètres	Code	Plage	Par défaut	Remarques
Adresse	<i>A d r e s s e</i>	0 - 31	1	Adresse du variateur,configurable par les commutateurs 3 à 7 (uniquement en lecture sur le terminal)
Protocole	<i>P r o t o c o l e</i>	UNI-TELWAY Modbus / RTU Modbus / ASCII	-	
Vitesse de transmission	<i>V i t e s s e</i>	4.8 9.6 19.2		Valeur en kbit / s
Format de trame (1)	<i>F o r m a t</i>	7o1 - 7E1 7o2 - 7E2 8o1		Pour Modbus / ASCII seulement
		8E1 - 8n1 8n2		Pour Modbus / ASCII et Modbus / RTU seulement
		8o1		Pour tous les protocoles

(1) Signification : 7 = 7 bits, 8 = 8 bits, o = impair, E = pair, n = sans parité, 1-2 = nombre de bits "stop".

# Principe de la communication

---

## Structure des données

Le réglage, la commande, le contrôle et la surveillance de l'Altivar 58 s'effectuent par l'intermédiaire de données (ou objets) qui sont propres à ce produit.

Ceux-ci sont constitués essentiellement de :

MOTS (de 16 bits) : désignés  $W_i$  ( $i$  = numéro du mot) qui seront utilisés pour mémoriser, soit des valeurs numériques entières (- 32768 à + 32767), soit 16 états logiques indépendants (ces mots sont alors appelés registres).

Exemples :

W401 = consigne de fréquence (valeur numérique),  
W483 = registre de défaut (16 bits de défaut).

Notation : W483,2 désigne le bit de rang 2 du registre W483.

## Accès aux données

Certaines données sont accessibles aussi bien en écriture qu'en lecture : ce sont les bits et les mots correspondant à des réglages, des consignes et des commandes. Ces données sont exploitées par le variateur.

Inversement, les données élaborées par le variateur ne sont accessibles qu'en lecture : informations de signalisation, de défaut,... . Leur écriture n'a pas de sens et est refusée.

**Les variables de l'Altivar 58 ainsi que son processus de commande par liaison série sont détaillés dans le guide d'exploitation "Variables internes de communication", à consulter.**

# Requêtes UNI-TELWAY

## Généralités

L'échange de données entre systèmes informatiques, automates programmables et autres systèmes intelligents doit s'effectuer dans un langage commun.

Ce langage doit être le plus simple possible et compris par chaque interlocuteur, néanmoins chaque échange doit pouvoir être contrôlé afin d'assurer l'intégrité des transferts. Les variables échangées sont alors insérées dans une trame constituée généralement de la façon suivante :



Chaque protocole définit la présence, le format, le contenu des différents groupes de variables entourant la zone de données.

Cette structuration permet de définir le début des messages, la taille de ceux-ci, éventuellement le système auquel sont adressées les données, le type de fonction demandée, les variables elles-mêmes, un paramètre de contrôle et un code de fin validant l'ensemble du message. Cette trame est différente par son contenu et sa forme pour chaque type de protocole.

## Liste des requêtes

Le tableau ci-après précise les requêtes acceptées par l'Altivar 58 et leurs limites. Le détail du codage des requêtes est donné dans le manuel de référence UNI-TELWAY.

Requête	Code (hexa)	Altivar 58
Identification	H'0F'	Oui
Version protocole	H'30'	Oui
Miroir	H'FA'	Oui
Lecture compteurs d'erreurs	H'A2'	Oui
RAZ compteurs	H'A4'	Oui
Lecture d'un mot	H'04'	Oui
Ecriture d'un mot	H'14'	Oui
Lecture d'objets	H'36'	63 mots maxi
Ecriture d'objets	H'37'	60 mots maxi
Spécifique	H'F2'	Voir détail

### Requête identification - Code requête H'0F'

Réponse fournie par l'Altivar

Code réponse = H'3F'

Type produit = H'18' pour Altivar

Sous-type = H'58' Altivar 58

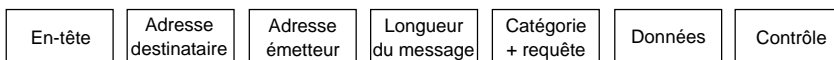
Version produit= H'XX' version logicielle (ex : H'21' pour V2.1)

Chaîne ASCII\* = calibre du variateur (ex : ATV58U18N4)

\* Le premier octet d'une chaîne ASCII correspond toujours à la longueur de la chaîne.

# Requêtes UNI-TELWAY

## Trame UNI-TELWAY



### Requêtes lecture et écriture d'objets

Ces requêtes permettent d'accéder à plusieurs mots dans les limites spécifiées à la page précédente. Le codage de ces requêtes peut s'effectuer en spécifiant :

Code question (TXTi,C) = H'36' (lecture) ou H'37' (écriture)  
 Catégorie = 0...7  
 Segment = H'68' (mot interne)  
 Type objet = H'07' pour mot (16 bits) en lecture et en écriture  
 Adresse objet = H'xxxx'  
 Etc...

Les mots réservés ou inutilisés sont lus à 0 et leur écriture est sans effet.  
 La réponse à la requête "écriture d'objets" est acceptée si au moins un mot est écrit.

**Exemple** : programmation sur automate TSX7 avec utilisation du bloc texte.  
 LECTURE des mots W250 à W253 de l'Altivar 58.

#### - En utilisant le type objet mot = H'07'

Bloc texte à l'émission

TxTi,C = H'0736' (catégorie + requête)

TxTi,L = 6

+ table d'émission

Bloc texte à la réception

TxTi,V = H'66' (compte-rendu)

TxTi,S = 9 (9 octets reçus)

+ table de réception

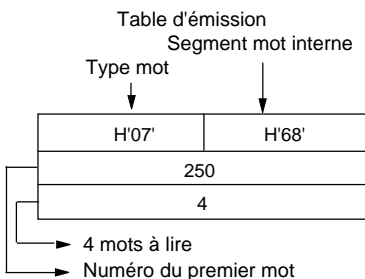


Table de réception

W250 (p. faible)	H'07'
W251 (p. faible)	W250 (p. fort)
W252 (p. faible)	W251 (p. fort)
W253 (p. faible)	W252 (p. fort)
	W253 (p. fort)

Les données reçues dans la table de réception sont décalées d'un octet. Il appartient au programme d'application d'en effectuer un recalage (par décalages successifs par exemple) avant de les exploiter.

# Requêtes UNI-TELWAY

---

## Requête spécifique de commande

Cette requête permet d'effectuer la commande de l'Altivar 58 et d'obtenir en retour des informations essentielles au contrôle du variateur.

### Format de la requête

Code requête	: octet	= H'F2'
Catégorie	: octet	= 0...7
Code requête spécifique	: octet	= 0
Réservé	: octet	= 0
Commande	: mot	= CMD
Consigne	: mot	= FRH
Accélération	: mot	= ACC
Décélération	: mot	= DEC

### Format du compte-rendu

Code réponse	: octet	= H'F2'
Code réponse spécifique	: octet	= H'30'
Réservé	: octet	= 0
Consigne	: mot	= FRH
Registre d'état	: mot	= ETA
Registre de défaut	: mot	= FLT
Courant moteur	: mot	= LCR

### Réponse négative

Code réponse	: octet	= H'FD'
Cause	: nombre de paramètres incorrect	

## Trames Modbus

**Nota** : Dans la suite de ce document les fonctionnalités Modbus et Jbus sont regroupées sous le terme de Modbus.

Deux modes de transmission sont utilisables, un seul d'entre eux étant employé dans un système.

### Mode RTU

La trame définie pour le protocole Modbus ne comporte ni octets d'en-tête de message, ni octets de fin de message. Sa définition est la suivante :



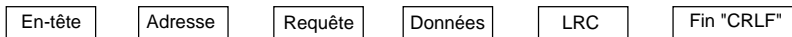
Les données sont transmises en binaire.

CRC16 : paramètre de contrôle polynomial (cyclical redundancy check).

La détection de fin de trame est réalisée sur un silence  $\geq 3$  caractères.

### Mode ASCII

La trame est complète et se définit de la façon suivante :



- en-tête = ":" (H'3A),
- les données sont codées en ASCII : chaque octet est divisé en 2 quartets et chaque quartet est codé par un caractère ASCII (0 à F),
- LRC : paramètre de contrôle longitudinal (longitudinal redundancy check),
- fin : "CR" "LF" (H'0D et H'0A).

# Protocole Modbus / Jbus

## Principe

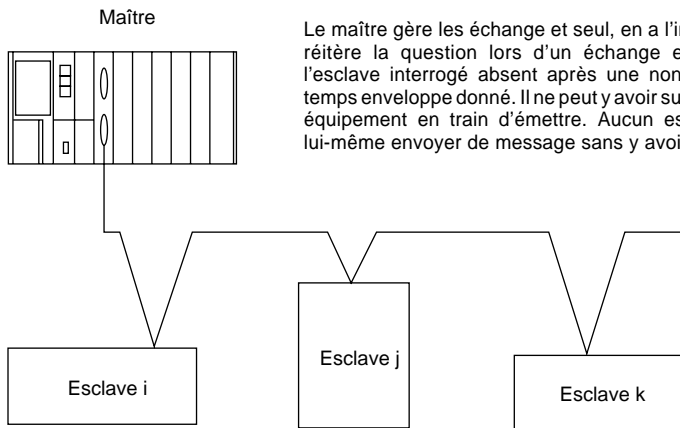
Le protocole Modbus est un protocole de dialogue créant une structure hiérarchisée (un maître et plusieurs esclaves).

Le protocole Modbus permet d'interroger depuis le maître, un ou plusieurs esclaves intelligents. Une liaison multipoint relie entre eux maître et esclaves.

Deux types de dialogue sont possibles entre maître et esclaves :

- le maître parle à un esclave et attend sa réponse,
- le maître parle à l'ensemble des esclaves sans attendre de réponse (principe de la diffusion générale).

Le numéro d'esclave varie de 1 à 255 et le numéro 0 est réservé pour la diffusion.



## Nota

Toute communication latérale (c'est-à-dire d'esclave à esclave) ne peut s'effectuer directement. Il est nécessaire que le logiciel d'application du maître ait été conçu en conséquence : interroger un esclave et renvoyer les données reçues à l'autre esclave.



# Protocole Modbus / Jbus

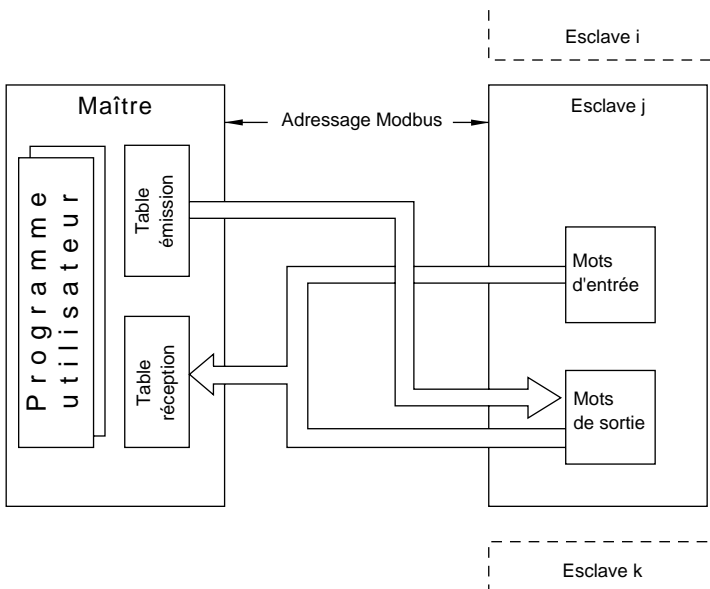
## Données accessibles

Le protocole Modbus permet d'échanger des données entre un maître et des esclaves, et assure le contrôle des échanges.

Par conséquent, dans chaque entité d'esclave, sont définies des zones de mots qui seront lues ou écrites par le maître.

Un objet d'entrée peut être lu uniquement.

Un objet de sortie peut être lu ou écrit.



## Les échanges

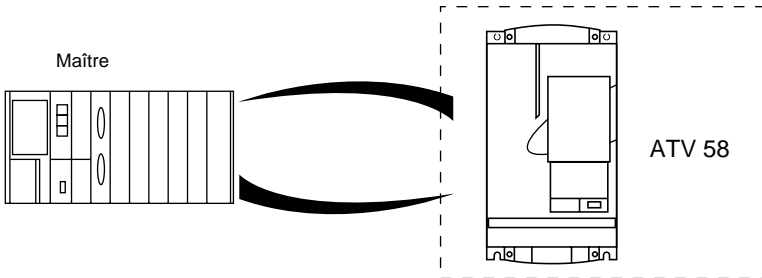
Le maître, ou organe de supervision, a l'initiative des échanges. Ce maître va s'adresser à un esclave en lui fournissant quatre types d'informations :

- adresse de l'esclave,
- fonction demandée à l'esclave,
- zone de données (variable en fonction de la requête),
- contrôle d'échange.

Le maître de la liaison attend la réponse de l'esclave avant d'émettre le message suivant, évitant ainsi tout conflit sur la ligne. Ceci autorise donc un fonctionnement en half-duplex.

## Contrôle et surveillance

Toute gestion d'échanges entre deux entités dialoguant par liaison série asynchrone, inclut évidemment des réponses d'exception lorsque sont apparus des défauts d'échange. Différents messages incohérents peuvent arriver à un esclave. Dans ce cas, ce dernier répond sa non-compréhension au maître qui prend ou non la décision de réitérer l'échange.



Le maître a accès à un certain nombre d'informations détenues et gérées par l'esclave. Le maître accède à ces données par des codes fonctions particuliers (mode diagnostic, lecture du compteur d'évènements,...).

## Fonctions Modbus

Parmi les fonctions Modbus on distingue :

- les fonctions principales permettant l'échange des données,
- les fonctions complémentaires pour le diagnostic des échanges.

Le tableau suivant indique les fonctions gérées par l'option communication de l'ALTIVAR 58, et précise en outre les limites.

La définition des fonctions "lecture" et "écriture" s'entend vue du maître.

Code	Nature des fonctions	D	ALTIVAR 58
03	Lecture de N mots de sortie		63 maxi
04	Lecture de N mots d'entrée		63 maxi
06	Ecriture d'un mot de sortie	D	Oui
08	Diagnostic (voir détails)		Oui
11	Lecture compteur d'événements		Oui
16	Ecriture de N mots de sortie	D	60 maxi

**Les fonctions notées «D» peuvent être utilisées en diffusion générale.**

**Le message émis par le maître doit alors spécifier un numéro d'esclave = 0.**

**Il n'y a jamais de message réponse en retour.**

## Détails des fonctions

Code 03 : lecture de N mots de sortie.

Cette fonction permet la lecture de mots de sortie (mots qui peuvent être écrits et lus par le maître dans l'esclave).

Code 04 : lecture de N mots d'entrée. Idem précédemment, mais s'applique aux mots d'entrée (mots que le maître ne peut que lire).

Code 06 : écriture d'un mot de sortie.

Permet d'effectuer l'écriture d'un mot de sortie de 16 bits (seuls accessibles en écriture).

# Protocole Modbus / Jbus

---

Le code fonction diagnostic 08 est toujours accompagné d'un sous-code.

Code 08/00 : écho.

Cette fonction demande à l'esclave interrogé de retourner intégralement le message envoyé par le maître.

Code 08/01 : réinitialisation de la voie.

Cette fonction permet de réinitialiser la communication d'un esclave et en particulier de lui faire quitter le mode écoute seule (LOM) par l'envoi d'une donnée H'0000 ou H'FF00.

Code 08/03 : changement délimiteur ASCII.

En mode ASCII, les messages sont délimités par le caractère line feed (LF = H'0A). Cette fonction permet de changer ce caractère.

Code 08/04 : passage en mode LOM.

Cette fonction permet de forcer un esclave à passer en écoute seule (LOM). Dans ce mode l'esclave ne traite pas les messages qui lui sont adressés, et n'émet jamais de réponse à l'exception de la réinitialisation de la voie.

Code 08/0A : remise à zéro des compteurs.

Cette fonction effectue la remise à zéro de tous les compteurs de surveillance des échanges d'un esclave.

Code 08/0B : nombre de messages corrects vus sur la ligne sans erreur CRC ou checksum. Cette fonction permet de lire sur un compteur 16 bits (incrémenté de 0 à H'FFFF) qui totalise les messages vus sur la ligne et traités par l'esclave.

Code 08/0C : nombre de messages reçus avec erreur de checksum (lecture d'un compteur de 16 bits).

Code 08/0D : nombre de réponses d'exception.

Lecture d'un compteur 16 bits totalisant le nombre de messages d'exception émis par un esclave vers le maître (suite à une trame incorrecte).

Code 08/0E : nombre de messages adressés à l'esclave sauf en diffusion.

Lecture d'un compteur 16 bits totalisant tous les messages adressés à l'esclave quelle que soit leur nature.

Code 08/0F : nombre de messages de diffusion reçus.

Lecture d'un compteur 16 bits totalisant tous les messages adressés à l'esclave quelle que soit leur nature.

Code 08/10 : lecture du nombre de réponses NAQ. La valeur lue est toujours 0.

Code 08/11 : lecture du nombre de réponses de l'esclave non prêt. La valeur lue est toujours 0.

Code 08/12 : lecture du nombre de caractères non traités (erronnés).

Code 11 : lecture compteur d'évènements.

– un status (toujours nul),  
– un compteur qui est incrémenté à chaque réception de message correct (forme et contenu) destiné à l'esclave sauf pour les réponses d'exception.

Code 16 : écriture de N mots de sortie.

Cette fonction permet au maître d'écrire des mots de sortie dans l'esclave (mots pouvant être écrits ou lus).

# Protocole Modbus / Jbus

## Détails des trames (mode RTU)

### Calcul du CRC16

Le CRC16 se calcule sur tous les octets du message en appliquant la méthode suivante.

Initialiser le CRC (registre de 16 bits) à H'FFFF.

Faire du 1er octet du message au dernier :

CRC XOR <octet> → CRC

Faire 8 fois

Décaler le CRC d'un bit à droite

Si le bit sorti = 1, faire CRC XOR H'A001 → CRC

Fin faire

Fin faire

Le CRC obtenu sera émis poids faibles d'abord, poids forts ensuite.

XOR = OU exclusif.

### Lecture de N mots : fonction 3 ou 4

Question

N° esclave	03 ou 04	N° du 1er mot PF   Pf	Nombre de mots PF   Pf	CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Réponse

N° esclave	03 ou 04	Nombre d'octets lus	Valeur 1er mot PF   Pf	.....	Valeur du dernier mot PF   Pf	CRC16
1 octet	1 octet	1 octet	2 octets		2 octets	2 octets

Exemple : lecture des mots W463 à W466 de l'esclave 1

Question

01	03	01CF	0004	75CA
----	----	------	------	------

Réponse

01	03	08	xxxx	.....	xxxx	CRC16
----	----	----	------	-------	------	-------

Valeur de W463
Valeur de W466

### Écriture d'un mot de sortie : fonction 6

Question

N° esclave	06 PF	Numéro du mot Pf   PF	Valeur du mot Pf   PF	CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Réponse

N° esclave	06 PF	Numéro du mot Pf   PF	Valeur du mot Pf   PF	CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Exemple : écriture de la valeur H'0315' = 789 dans le mot W252 de l'esclave 1 ( ACC = 78,9s )

Question et réponse

01	06	00FC	0315	88C5
----	----	------	------	------

# Protocole Modbus / Jbus

## Diagnostic : fonction 8

Question et réponse

N° esclave	08	Sous-code	Données	CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Sous-code	Données questions	Données réponses	Fonction exécutée
00	XX YY	XX YY	Echo
01	00 00	00 00	Réinitialisation
03	XX 00	XX 00	XX = nouveau délimiteur
04	00 00	Pas de réponse	Passage en mode LOM
0A	00 00	00 00	Remise à 0 compteurs
0B	00 00	XX YY	XXYY = valeur compteur
0C	00 00	XX YY	XXYY = valeur compteur
0D	00 00	XX YY	XXYY = valeur compteur
0E	00 00	XX YY	XXYY = valeur compteur

## Lecture du compteur d'événements : fonction 11 ( 'H'0B' )

Question

N° esclave	0B	CRC16
1 octet	1 octet	2 octets

Réponse

N° esclave	0B	00   00	Valeur compteur PF   Pf	CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

## Ecriture de N mots de sortie : fonction 16 ( 'H'10' )

Question

N° esclave	10	N° du 1er mot PF   Pf	Nombre de mots	Nombre d'octets	Valeur 1er mot PF   Pf	...	CRC16
1 octet	1 octet	2 octets	2 octets	1 octet	2 octets		2 octets

Réponse

N° esclave	10	N° 1er mot PF   Pf	Nombre de mots PF   Pf	CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

Exemple : écriture des valeurs 6 et 500 dans les mots W400 et W401 de l'esclave 2

Question	02	10	0190	0002	04	0006	01F4	1801
----------	----	----	------	------	----	------	------	------

Réponse	02	10	0190	0002	402A
---------	----	----	------	------	------

# Protocole Modbus / Jbus

---

## Réponses d'exception

Une réponse d'exception est retournée par un esclave lorsque celui-ci ne peut exécuter la demande qui lui est adressée.

Format d'une réponse d'exception :

N° esclave	Code réponse	Code erreur	CRC16
1 octet	1 octet	1 octet	2 octets

**Code réponse** : code fonction de la demande + H'80 (le bit de rang le plus élevé est mis à 1).

**Code erreur** : 1 = la fonction demandée n'est pas reconnue par l'esclave.

2 = les numéros (adresses) de bits et de mots indiqués lors de la demande n'existent pas dans l'esclave.

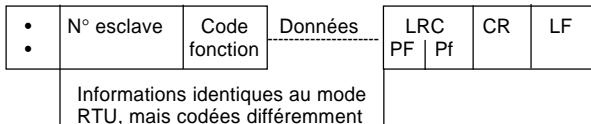
3 = les valeurs de bits et de mots indiquées lors de la demande ne sont pas permises dans l'esclave.

4 = l'esclave a commencé à exécuter la demande, mais ne peut continuer à la traiter entièrement.

# Protocole Modbus / Jbus

## Mode ASCII

Dans ce mode, la trame Modbus possède la structure suivante :



**Délimiteurs** : ":" = H'3A', CR = H'0D', LF = H'0A'.

**Informations** : le champ information est analogue aux trames RTU, mais codé en caractères ASCII. Chaque octet est séparé en 2 quartets et chacun d'eux est codé par son équivalent ASCII.

Exemple : l'octet contenant le numéro d'esclave 06 sera codé par les 2 caractères ASCII "0" et "6", c'est-à-dire par H'30' et H'36'.

**LRC** : somme hexadécimale modulo 256 du contenu de la trame (sans les délimiteurs) avant codage en ASCII, complémentée à 2.

L'octet obtenu est ensuite codé sous forme de 2 caractères ASCII comme précédemment.

Exemple : écriture de la valeur 10 dans le mot W252 de l'esclave 2

Question et réponse

En ASCII

3A	30 32	30 36	30304643	30303041	4632	0D	0A
----	-------	-------	----------	----------	------	----	----

En hexadécimal

:	02	06	00FC	000A	F2	CR	LF
---	----	----	------	------	----	----	----

Calcul du LRC

Somme des octets de la trame :

$$H'02' + H'06' + H'00' + H'FC' + H'00' + H'0A' = H'10E' = 270$$

Somme modulo 256 : H'0E' = 14

Complément à 2 de la somme modulo 256 :

$$H'100' - H'0E' = 256 - 14 = 242 = H'F2'$$



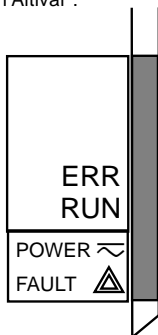
# Diagnostic

## Défaut

Pour la signification des codes, se reporter aux pages indiquant les mots de signalisation (guide d'exploitation "Variables internes de communication").

## Diagnostic complémentaire

Vérifier l'état des 2 voyants RUN et ERR situés sur la carte et visibles au travers du capot de l'Altivar :



ERR : voyant rouge  
RUN : voyant vert

Etats des voyants :  
0 = éteint      1/2 = clignotant lent (500 ms)  
1 = allumé      1/10 = clignotant rapide (100 ms)

Voyant RUN vert	Voyant ERR rouge	Causes probables	Actions correctives
1	0	Fonctionnement normal, bus et variateur présents	OK
0	0	Hors service, hors tension	Vérifier l'alimentation
0	1	Défaut de communication sur le bus	Vérifier le bus de communication et la connectique. Vérifier les commutateurs des prises abonnées
0	1/10 *	Erreur caractère	Vérifier la configuration de la communication
1/10	0	Communication non configurée	Configurer la communication
1/2	0	Défaut communication entre la carte de communication et l'Altivar	Vérifier le connecteur 30 points entre la carte et l'Altivar

\* Cet affichage est un clignotement d'une durée de 6 x 100 ms si un caractère erroné a été reçu. Ce clignotement court est répété après une période de cinq secondes d'extinction si un caractère reçu est erroné. Il n'a lieu que lorsque la carte communication est en défaut communication (pas de message reçu depuis 10 s en communication bus ou 1 s en protocole ASCII).

Dans le cas où la carte ne passe jamais en fonctionnement normal, cet affichage indique que le câblage est certainement correct mais que la configuration est inadaptée (vitesse ou format).





When the speed controller is powered up, the power components and some of the control components are connected to the mains supply. *It is extremely dangerous to touch them. The speed controller cover must be kept closed.*

After switching the power to the ALTIVAR off, *wait for 3 minutes before working on the equipment.* This is the time required for the capacitors to discharge.

Although every care has been taken in the preparation of this document, Schneider Electric SA cannot guarantee the contents and cannot be held responsible for any errors it may contain nor for any damage which may result from its use or application.

The products and options described in this document may be changed or modified at any time, either from a technical point of view or in the way they are operated. Their description can in no way be considered contractual.

# Contents

---

<b>Hardware Setup</b>	<b>28</b>
Introduction	28
Installing the Card, Address	29
<b>Connection to the multidrop bus</b>	<b>31</b>
<b>Software Setup</b>	<b>33</b>
Configuring Communication Functions	33
<b>Communication principle</b>	<b>34</b>
<b>UNI-TELWAY requests</b>	<b>35</b>
UNI-TELWAY frame	36
Specific control request	37
<b>Modbus / Jbus protocol</b>	<b>38</b>
Modbus frames	38
Details of frames (RTU mode)	44
ASCII mode	47
<b>Diagnostics</b>	<b>48</b>

---

# Hardware Setup

---

## Introduction

The VW3-A58303 communication card is used to connect an Altivar® 58 speed controller to UNI-TELWAY et Modbus / Jbus networks.

It is delivered with a 3 meter connection cable equipped with two connectors :

- a 9 pin Sub D connector for the card,
- a 16 pin Sub D connector for the bus.

The Altivar 58 speed controller can receive and respond to data messages. This data exchange enables a network to access certain Altivar 58 functions such as :

- Remote loading of configuration parameters et de réglages,
- Command and Control,
- Monitoring,
- Diagnostics.

# Hardware Setup

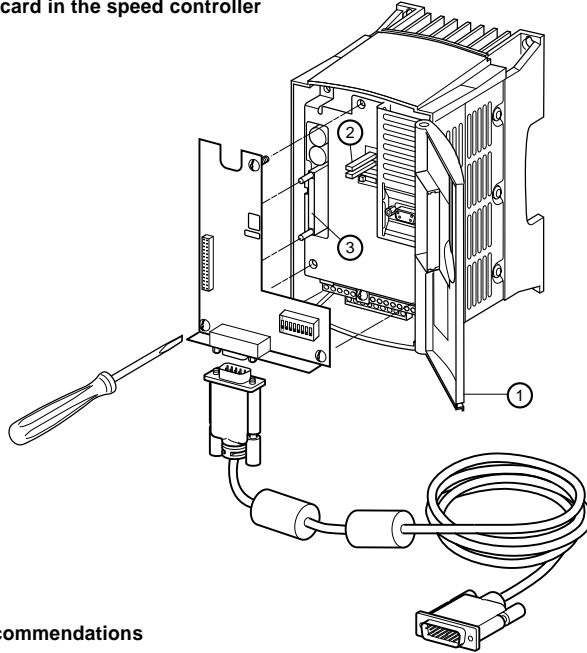
## Installing the Card, Address

### Acceptance

Ensure that the card reference printed on the label is the same as that on the delivery note corresponding to the purchase order.

Remove the option card from its packaging and check that it has not been damaged in transit.

### Installing the card in the speed controller



### Mounting Recommendations

Ensure that the speed controller is powered down.

To access the mounting slot for the option card, unlock cover (1) and open it.

Check that there is no power to the DC bus. Green LED (2) (POWER) must be off : wait 3 minutes after powering down.

Open the flexible protection of the control card.

Mount the option card on the control card support by snapping it onto connector (3) and fix it by means of the three screws.

The cable is connected to the card by a 9 pin connector.

### Put in place the label

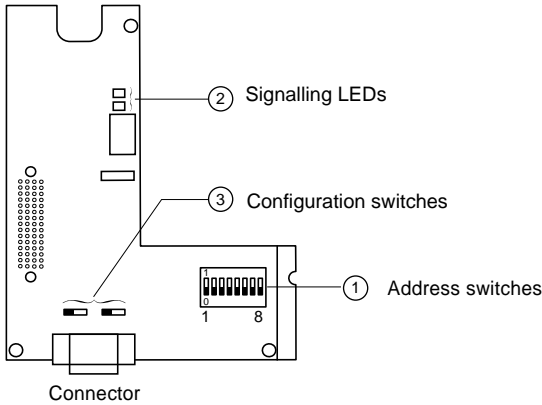


The card is supplied with a self-adhesive label indicating the functions of the signalling LEDs.

Stick the label on the cover of the Altivar 58 below the existing label.

# Hardware Setup

## Card layout :



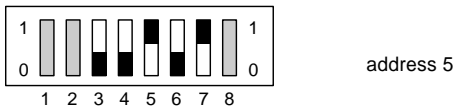
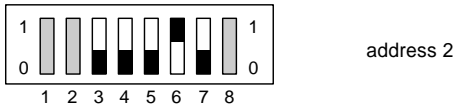
## Coding the speed controller address :

An Altivar 58 is identified on the bus by its address, coded from 0 to 31.

The address corresponds to the binary number given by setting the 8 switches (1) on the card to 1 or 0 (in fact only the switches 3 to 7 are used for the address).

**Caution :** the least significant bits are on the right.

Examples :



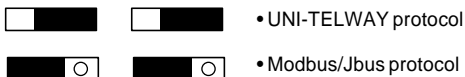
Setting all switches to 0 gives the address 0.

## Signalling LED :

The UNI-TELWAY, Modbus/Jbus card has 2 lamps (2) whose function is detailed in the diagnostic chapter.

## Configuration switches :

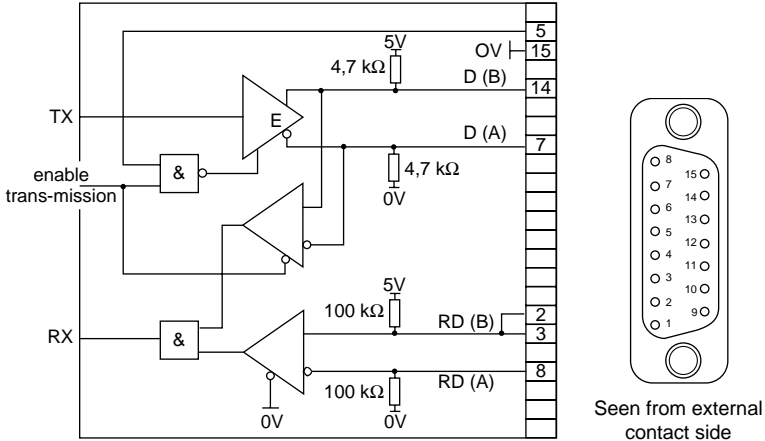
The card has two switches (3) for the protocol configuration.



# Connection to the multidrop bus

## SUB-D connector pinout

The RS 485 standard transmission interface is electrically isolated from the speed controller. It is available on a 15-pin SUB-D connector.

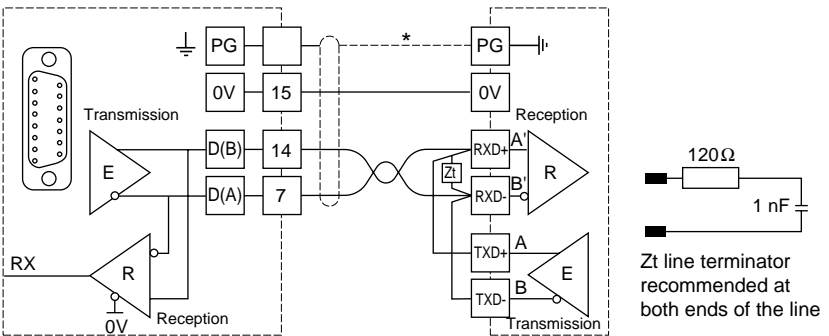


## Connection to the standard RS 485 bus

### Pins to use

SUB-D connector (15-pin)

Automated system



\* Connection of the shielding to both ends depends on installation constraints

## Recommendations

- use a shielded cable with 2 pairs of twisted conductors,
- connect the reference potentials to each other,
- maximum length of the line : 1000 meters,
- maximum length of a tap-link : 20 meters,
- do not connect more than 28 stations on a bus,
- cable routing : keep the bus away from the power cables (30 cm minimum), and make right-angle crossovers if necessary; connect the cable shielding to the earth of each device,
- fit a line terminator at both ends of the line.



# Connection to the multidrop bus

The following accessories are available for connecting equipment.

**TSX-CSA...** cables for bus sold in 100, 200 or 500 m lengths.

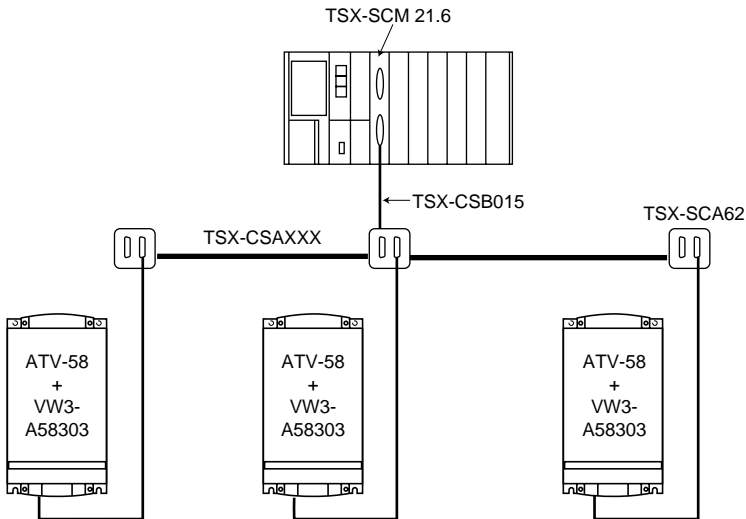
**TSX-SCA62** subscriber connector

This passive unit comprises a printed circuit fitted with screw terminals for connecting 2 devices to the bus. It includes an end of line terminator when the connector is located at the end of the bus. The switches on the connector must be set in the following way.

switch number	switch position
2	OFF
3	OFF
5	OFF

the position of the other switches has no effect.

## Example of connection to a UNI-TELWAY bus



# Software Setup

## Configuring Communication Functions

### Initial power-up

When the Altivar 58 is connected to the supply the card is automatically recognised. The configuration menu 8-COMMUNICATION will appear on the display terminal (or programming terminal or PC software).

### Configuration

Select the menu 8-COMMUNICATION for access to the configuration parameters of the card. This menu enables configuration of all the communication parameters.

The first parameter is the network address of the drive, this parameter is only accessible on the terminal, because it is configured by switches 3 to 7. The second parameter is the choice of protocol.

Modification of the communication parameters is only possible when the motor has stopped.

Parameter	Code	Range	Factory setting	Comments
Address	<i>A d r C</i>	0 - 31	1	Drive address configurable by switches 3 to 7 (reading only)
Protocol	<i>- P r o</i>	UNI-TELWAY Modbus / RTU Modbus / ASCII	-	
Transmission speed	<i>- b d r</i>	4.8 9.6 19.2		Value in Kbit/s
Format (1)	<i>- F o r</i>	7o1 - 7E1 7o2 - 7E2 8o1		For Modbus ASCII only
		8E1 - 8n1 8n2		For Modbus / ASCII and Modbus / RTU only
		8o1		For all protocols

(1) Signification : 7 = 7 bits, 8 = 8 bits, o = uneven, E = even, n = without parity bit, 1-2 = number of "stop" bits.

# Communication principle

---

## Data structure

The adjustment, control, supervision and monitoring of the Altivar 58 are performed using data (or objects) which are specific to the product.

The data essentially comprises :

WORDS (of 16 bits) : named  $W_i$  ( $i$  = word number) which are used for storage, either of complete digital values (- 32768 to + 32767), or of 16 independent logic states (in which case these words are termed registers).

Examples :

W401 = frequency reference (digital value),  
W483 = fault register (16 fault bits).

Notation : W483,2 designates the bit in row 2 of register W483.

## Access to data

Some data can be accessed in both read and write : these are the bits and words corresponding to adjustments, references and commands. This data is used by the speed controller.

However, data produced by the speed controller can only be accessed in read : signalling or fault data, etc. If written, they have no meaning and are rejected.

**The variables of the ALTIVAR 58 and their control procedure are detailed in the manual "Internal communication variables".**

# UNI-TELWAY requests

## General

The exchange of data between computer systems, PLCs and other intelligent systems must be performed using a common language.

This language should be as simple as possible and understood by everyone involved. Nevertheless, it must be possible to check every exchange to ensure the integrity of the transfers. The variables exchanged are therefore inserted in a frame which generally comprises the following :



Each protocol defines the presence, the format and the contents of the various groups of variables which surround the data zone.

This structuring makes it possible to define the start and the size of messages, if necessary the system to which the data is addressed, the type of function required, the variables themselves, a control parameter and an end code which validates the whole message.

The form and content of this frame are different for each type of protocol.

## List of requests

The following table describes the requests accepted by the Altivar 16 and their limits. Details of the coding of the requests are given in the UNI-TELWAY reference manual.

Request	Code (hexa)	Altivar 58
Identification	H'0F'	Yes
Protocol version	H'30'	Yes
Mirror	H'FA'	Yes
Read error counters	H'A2'	Yes
Reset counters	H'A4'	Yes
Read a word	H'04'	Yes
Write a word	H'14'	Yes
Read objects	H'36'	63 words max.
Write objects	H'37'	60 words max.
Specific	H'F2'	See later

## Identification request - Request code H'0F'

Response given by Altivar

Response code = H'3F'

Product type = H'18' for Altivar

Sub-type = H'58' Altivar 58

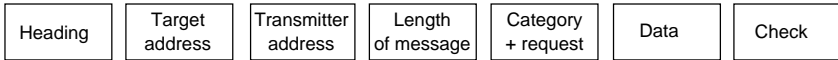
Product version = H'XX' software version (eg : H'21' for V2.1)

ASCII string\* = Altivar size (eg : ATV-58U18N4)

\* The first byte of an ASCII string always corresponds to the length of the string.

# UNI-TELWAY requests

## UNI-TELWAY frame



### Requests to read and write objects

These requests are used to access several words within the limits described on the previous page. These requests can be coded by specifying :

Question code (TxTi,C) = H'36' (read) or H'37' (write)  
 Category = 0...7  
 Segment = H'68' (internal word)  
 Object type = H'07' for a word (16 bits) in reading and writing  
 Object address = H'xxxx'  
 Etc...

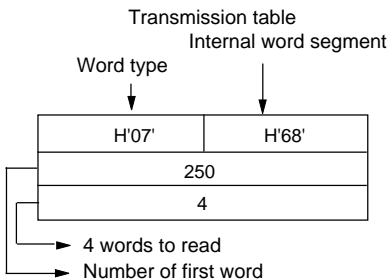
Words reserved or not used are read to 0 and their write is not significant. The response to a "write objects" request is accepted if a word is written at least.

**Example** : programming on a TSX7 PLC using a text block.  
 READ words W250 to W253 of the Altivar 58.

#### - Using word type object = H'07'

Transmission text block  
 TxTi,C = H'0736' (category + request)  
 TxTi,L = 6  
 + transmission table

Reception text block  
 TxTi,V = H'66' (confirm)  
 TxTi,S = 9 (9 bytes received)  
 + reception table



Reception table

W250 (least sig.)	H'07'
W251 (least sig.)	W250 (most sig.)
W252 (least sig.)	W251 (most sig.)
W253 (least sig.)	W252 (most sig.)
	W253 (most sig.)

The data received in the reception table is offset by one byte. It is the application program which must correct the data (for example by successive offsets) before using it.

# UNI-TELWAY requests

---

## Specific control request

This request is used to control the Altivar 58 and to obtain in return data essential for controlling the speed controller.

### Request format

Request code	: byte	= H'F2'
Category	: byte	= 0...7
Specific request code	: byte	= 0
Reserved	: byte	= 0
Command	: word	= CMD
Reference	: word	= FRH
Acceleration	: word	= ACC
Deceleration	: word	= DEC

### Confirm format

Response code	: byte	= H'F2'
Specific response code	: byte	= H'30'
Reserved	: byte	= 0
Reference	: word	= FRH
Status register	: word	= ETA
Fault register	: word	= FLT
Motor current	: word	= LCR

### Negative response

Response code	: byte	= H'FD'
Cause	:	incorrect number of parameters

# Modbus / Jbus protocol

---

## Modbus frames

Note : In the rest of this document Modbus and Jbus functions are grouped together under the heading Modbus.

Two transmission modes can be used, only one of them being used in a system.

### RTU mode

The frame defined for the Modbus protocol has neither message heading bytes nor end of message bytes. It is defined as follows :



The data is transmitted in binary code.

CRC16 : cyclical redundancy check.

The end of frame is detected on a silence of 3 characters or more.

### ASCII mode

The frame is complete and is defined in the following way :



- heading = ":" (H'3A),
- the data is coded in ASCII : each byte is divided into 2 four-bit bytes, each of which is coded by an ASCII character (0 to F),
- LRC : longitudinal redundancy check,
- end : "CR" "LF" (H'0D and H'0A).

# Modbus / Jbus protocol

## Principle

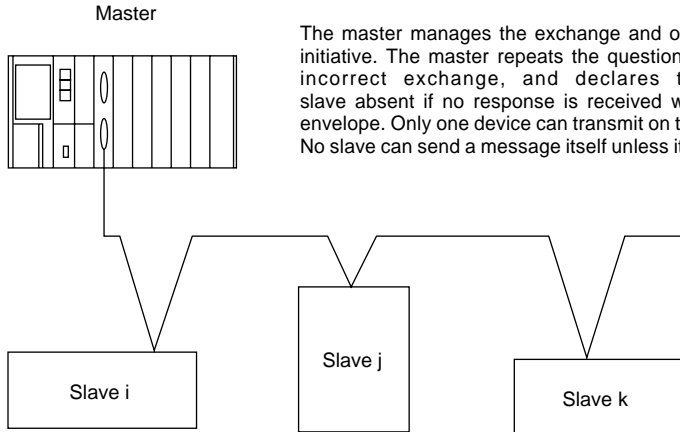
The Modbus protocol is a dialogue protocol which creates a hierarchical structure (a master and several slaves).

The Modbus protocol enables the master to interrogate one or more intelligent slaves. A multidrop link connects the master and slaves.

Two types of dialogue are possible between master and slaves :

- the master talks to a slave and waits for a response,
- the master talks to all the slaves without waiting for a response (broadcasting principle).

The slaves are numbered from 1 to 255, and number 0 is reserved for broadcasting.



The master manages the exchange and only it can take the initiative. The master repeats the question when there is an incorrect exchange, and declares the interrogated slave absent if no response is received within a given time envelope. Only one device can transmit on the line at any time. No slave can send a message itself unless it is invited to do so.

## Note

No lateral communication (ie. slave to slave) can be performed directly.

The application software of the master must therefore be designed to interrogate a slave and send back data received to another slave.



# Modbus / Jbus protocol

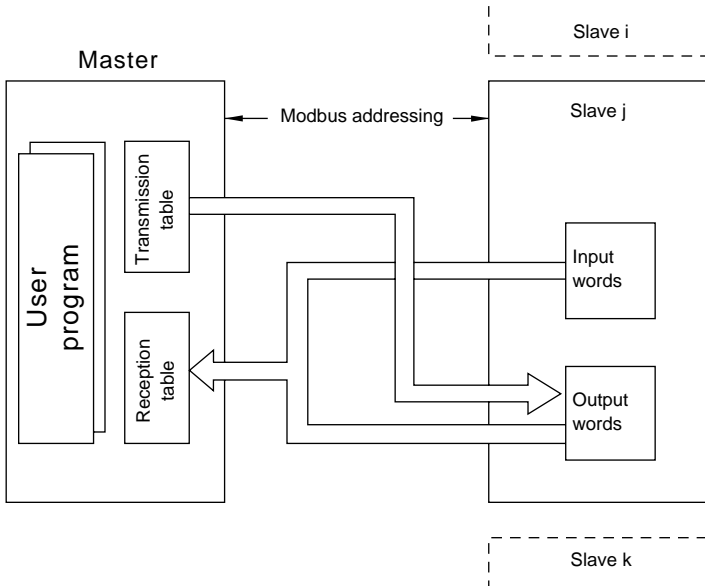
## Accessible data

The Modbus protocol enables data (words) to be exchanged between a master and several slaves, and checks these exchanges.

Consequently, word areas are defined in each slave unit which will be read or written by the master.

An input object can only be read.

An output object can be read or written.



## Exchanges

The master, or supervision device, takes the initiative in exchanges. The master addresses a slave by supplying it with four types of data :

- the address of the slave,
- the function required of the slave,
- the data area (variable depending on the request),
- the exchange check.

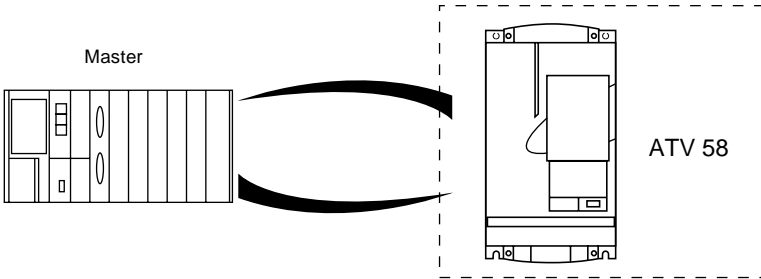
The link master waits for the response of the slave before transmitting the next message, thus avoiding any conflict on the line. Operation in half duplex is therefore authorized.

# Modbus / Jbus protocol

---

## Control and monitoring

All control of exchanges between two units which are communicating via asynchronous serial link naturally includes exception messages when exchange faults occur. Various incorrect messages may be sent to a slave. In this event, the slave will tell the master that it does not understand, and the master will decide whether or not to repeat the exchange.



The master has access to a certain amount of data which is stored and managed by the slave. The master can access this data using special function codes (diagnostic mode, read event counter, etc).

# Modbus / Jbus protocol

---

## Modbus functions

Modbus functions include :

- main functions for exchanging data,
- additional functions for exchange diagnostics.

The following table shows the functions which are managed by the ALTIVAR 58 communication function, and specifies its limits.

The definition of the "read" and "write" functions are understood from the point of view of the master.

Code	Type of function	D	ALTIVAR 58
03	Read N output words		63 max
04	Read N input words		63 max
06	Write one output word	D	Yes
08	Diagnostics (see details below)		Yes
11	Read event counter		Yes
16	Write N output words	D	60 max

**Functions marked «D» can be broadcast.**

**The message transmitted by the master must specify slave number = 0.**

**A response message is never returned.**

## Detailed information on functions

Code 03 : read N output words.

This function is used to read output words (words which can be written and read in the slave by the master ).

Code 04 : read N input words.

As above, but applies to input words (words which the master can only read).

Code 06 : write an output word

Used to write a 16-bit output word (can only be accessed in write).

# Modbus / Jbus protocol

---

Diagnostic function code 08 is always accompanied by a sub-code.

- Code 08/00 : echo.  
This function requests the interrogated slave to send back the whole message sent by the master.
- Code 08/01 : channel reinitialization.  
This function is used to reinitialize communication of a slave and in particular to make it leave listen only mode (LOM) by transmission of a data H'0000 ou H'FF00.
- Code 08/03 : change of ASCII delimiter.  
In ASCII mode, messages are delimited by the line feed character (LF = H'0A). This function is used to change this character.
- Code 08/04 : change to LOM mode.  
This function is used to force a slave into listen only mode (LOM). In this mode the slave does not process messages which are addressed to it, and only transmits a response when the channel is reinitialized.
- Code 08/0A : counter reset.  
This function resets to zero all the counters monitoring the exchanges of a slave.
- Code 08/0B : number of correct messages seen on the line without CRC error or checksum error. This function reads a 16-bit counter (incremented from 0 to H'FFFF) which totals the messages seen on the line and processed by the slave.
- Code 08/0C : number of messages received with checksum error (reads a 16-bit counter).
- Code 08/0D : number of exception responses.  
Reads a 16-bit counter which totals the number of exception messages transmitted to the master by a slave (following an incorrect frame).
- Code 08/0E : number of messages addressed to the slave except for broadcasts.  
Reads a 16-bit counter which totals the number of all types of messages addressed to the slave.
- Code 08/0F : number of broadcast messages received.  
Reads a 16-bit counter which totals the number of all types of messages addressed to the slave.
- Code 08/10 : read number of NAQ responses. The value read is always 0.
- Code 08/11 : read of number of responses from the slave that is not ready. The value read is always 0.
- Code 08/12 : read the number of characters which are not processed (incorrect).
- Code 11 : read event counter.  
– a status (always zero),  
– a counter which is incremented each time a correct message sent to the slave is received (form and content) except for exception messages.
- Code 16 : write N output words.  
This function enables the master to write output words to the slave (words which can be written or read).

# Modbus / Jbus protocol

## Details of frames (RTU mode)

### CRC16 calculation

The CRC16 is calculated based on all the bytes of the message by applying the following method. Initialize the CRC (16-bit register) to H'FFFF.

Enter the first to the last byte of the message :

CRC XOR <byte> → CRC

Enter 8 times

Move the CRC one bit to the right

If the output bit = 1, enter CRC XOR H'A001 → CRC

End enter

End enter

The CRC obtained will be transmitted least significant byte first, then most significant.

XOR = exclusive OR.

**Read N words** : function 3 or 4

Question

Slave n°	03 or 04	N° of 1st word		Number of words		CRC16
1 byte	1 byte	MS	LS	MS	LS	2 bytes
		2 bytes		2 bytes		

Réponse

Slave n°	03 or 04	Number of bytes read	Value of 1st word		Value of last word		CRC16
1 byte	1 byte	1 byte	MS	LS	MS	LS	2 bytes
		1 byte	2 bytes		2 bytes		2 bytes

Example : read words W463 to W466 of slave 1

Question	01	03	01CF	0004	75CA
----------	----	----	------	------	------

Response	01	03	08	xxxx	-----	xxxx	CRC16
				Value of W463		Value of W466	

**Write an output word** : function 6

Question

Slave n°	06	Word number		Word value		CRC16
1 byte	MS	LS	MS	LS		2 bytes
	1 byte	2 bytes		2 bytes		

Response

Slave n°	06	Word number		Word value		CRC16
1 byte	MS	LS	MS	LS		2 bytes
	1 byte	2 bytes		2 bytes		

Example : write value H'0315' = 789 in word W252 of slave 1 ( ACC = 78,9s )

Question and response	01	06	00FC	0315	88C5
-----------------------	----	----	------	------	------

ENGLISH

# Modbus / Jbus protocol

## Diagnostic : function 8

Question and response

Slave n°	08	Sub-code	Data	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Sub-code	Question data	Response data	Function executed
00	XX YY	XX YY	Echo
01	00 00	00 00	Reinitialization
03	XX 00	XX 00	XX = new delimiter
04	00 00	No response	Change to LOM mode
0A	00 00	00 00	Reset counters to 0
0B	00 00	XX YY	XXYY = counter value
0C	00 00	XX YY	XXYY = counter value
0D	00 00	XX YY	XXYY = counter value
0E	00 00	XX YY	XXYY = counter value

## Read event counter : function 11 ( 'H'0B' )

Question

Slave n°	0B	CRC16
1 byte	1 byte	2 bytes

Response

Slave n°	0B	00   00	Counter value MS   LS	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

## Write N output words : function 16 ( 'H'10' )

Question

Slave n°	10	N° of 1st word MS   LS	Number of words	Number of bytes	Value of 1st word MS   LS	... CRC16
1 byte	1 byte	2 bytes	2 bytes	1 byte	2 bytes	2 bytes

Response

Slave n°	10	N° of 1st word MS   LS	Number of words MS   LS	CRC16
1 byte	1 byte	2 bytes	2 bytes	2 bytes

Example : write values 6 and 500 in words W400 and W401 of slave 2.

Question	02	10	0190	0002	04	0006	01F4	1801
----------	----	----	------	------	----	------	------	------

Response	02	10	0190	0002	402A
----------	----	----	------	------	------

# Modbus / Jbus protocol

---

## Exception responses

An exception response is given by a slave when it is unable to perform the request which is addressed to it.

Format of an exception response :

Slave n°	Response code	Error code	CRC16
1 byte	1 byte	1 byte	2 bytes

**Response code** : function code of the request + H'80 (the most significant bit is set to 1).

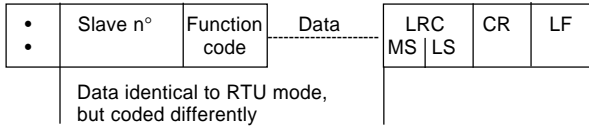
**Error code** :

- 1 = the function requested is not recognized by the slave.
- 2 = the bit and word numbers (addresses) indicated in the request do not exist in the slave.
- 3 = the bit and word values indicated in the request are not permissible in the slave.
- 4 = the slave has started to execute the request, but cannot continue to execute it completely.

# Modbus / Jbus protocol

## ASCII mode

In this mode, the Modbus frame has the following structure :



**Delimiters** : ":" = H'3A', CR = H'0D', LF = H'0A'.

**Data** : the data field is analogous to the RTU frames, but coded in ASCII characters. Each byte is divided into 2 four-bit bytes, each of which is coded by its ASCII equivalent.

Example : the byte containing the slave number 06 will be coded by 2 ASCII characters "0" and "6", ie. by H'30' and H'36'.

**LRC** : modulo 256 hexadecimal sum of the contents of the frame (without the delimiters) before ASCII coding, 2's complement.

The byte obtained is then coded in the form of 2 ASCII characters as above.

Example : write value 10 in word W252 of slave 2

Question and response

ASCII

3A	30 32	30 36	30304643	30303041	4632	0D	0A
----	-------	-------	----------	----------	------	----	----

Hexadecimal

:	02	06	00FC	000A	F2	CR	LF
---	----	----	------	------	----	----	----

LRC calculation

Sum of the bytes in the frame :

$$H'02' + H'06' + H'00' + H'FC' + H'00' + H'0A' = H'10E' = 270$$

$$\text{Modulo sum 256 : } H'0E' = 14$$

Modulo sum 256, 2's complement :

$$H'100' - H'0E' = 256 - 14 = 242 = H'F2'$$



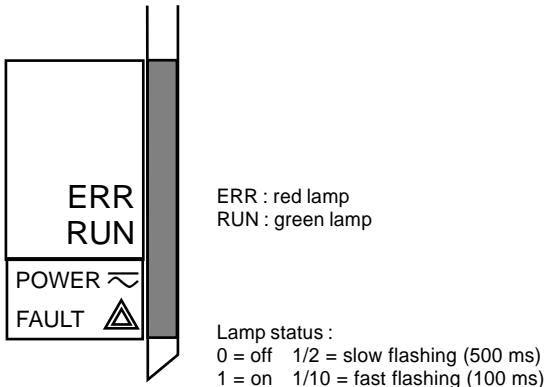
# Diagnostics

## Fault

For an explanation of the codes consult the manual «Internal communication variables».

## Additional diagnostics

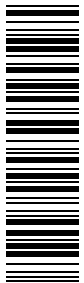
Check the state of the 2 lamps RUN and ERR mounted on the card and visible through the cover of the Altivar :



RUN lamp green	ERR lamp red	Probable cause	Corrective action
1	0	Normal operation, bus and starter present	OK
0	0	Not operating	Check supply
0	1	Communication fault on the bus	Check the communication bus and the connectors. Also check the switches
0	1/10 *	Character error	Check communication configuration
1/10	0	Communication not configured	Configure communication
1/2	0	Communication fault between the Altivar and communication card	Check 30 pin connector between Altivar and card

\* This display flashes for 6 x 100 ms if an incorrect character has been received. This short flashing is repeated after a period of five seconds if an incorrect character is received. This only occurs when the communication card is set for communication fault (no message received for 10 s for bus communication or 1 s for ASCII protocol).

If the card never changes to normal operation, this display indicates that the wiring is definitely correct but that the configuration is not suitable (speed or format).



0 33 89110 85593 7

VVDED397054

**85593**

W9 1598007 01 11 A02

**2001- 12**