Annexe 6 : Extrait documentation modbus

Cette annexe est un extrait du document « MODBUS over serial line specification and implementation guide V1.02 » disponible sur le site http://www.modbus.org.

2.5 The two serial Transmission Modes

Two different serial transmission modes are defined: The RTU mode and the ASCII mode.

It defines the bit contents of message fields transmitted serially on the line. It determines how information is packed into the message fields and decoded.

The transmission mode (and serial port parameters) must be the same for all devices on a MODBUS Serial Line.

Although the ASCII mode is required in some specific applications, interoperability between MODBUS devices can be reached only if each device has the same transmission mode: **All devices must implement the RTU Mode.** The ASCII transmission mode is an option.

Devices should be set up by the users to the desired transmission mode, RTU or ASCII. Default setup must be the RTU mode.

2.5.1 RTU Transmission Mode

When devices communicate on a MODBUS serial line using the RTU (Remote Terminal Unit) mode, each 8–bit byte in a message contains two 4–bit hexadecimal characters. The main advantage of this mode is that its greater character density allows better data throughput than ASCII mode for the same baud rate. Each message must be transmitted in a continuous stream of characters.

The format (11 bits) for each byte in RTU mode is:

Coding System: 8–bit binary

Bits per Byte: 1 start bit
8 data bits, least significant bit sent first
1 bit for parity completion
1 stop bit

Even parity is required; other modes (odd parity, no parity) may also be used. In order to ensure a maximum compatibility with other products, it is recommended to support also No parity mode. The default parity mode must be even parity.

Remark: the use of no parity requires 2 stop bits.

How Characters are Transmitted Serially:

Each character or byte is sent in this order (left to right):
Least Significant Bit (LSB) . . . Most Significant Bit (MSB)

Figure 10: Bit Sequence in RTU mode

Devices may accept by configuration either Even, Odd, or No Parity checking. If No Parity is implemented, an additional stop bit is transmitted to fill out the character frame to a full 11-bit asynchronous character:

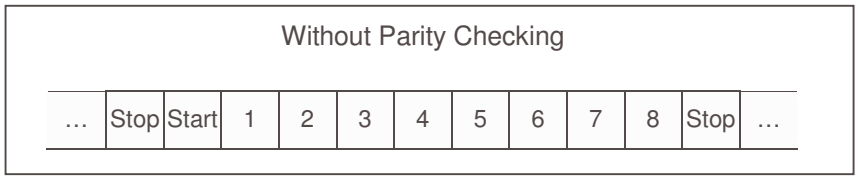


Figure 11: Bit Sequence in RTU mode (specific case of No Parity)

Frame Checking Field: Cyclical Redundancy Checking (CRC)

Frame description:

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low | CRC Hi

Figure 12: RTU Message Frame

➔ The maximum size of a MQDBUS RTU frame is 256 bytes.

2.5.1.1 MODBUS Message RTU Framing

A MODBUS message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows devices that receive a new frame to begin at the start of the message, and to know when the message is completed. Partial messages must be detected and errors must be set as a result.

In RTU mode, message frames are separated by a silent interval of at least 3.5 character times. In the following sections, this time interval is called t3,5.

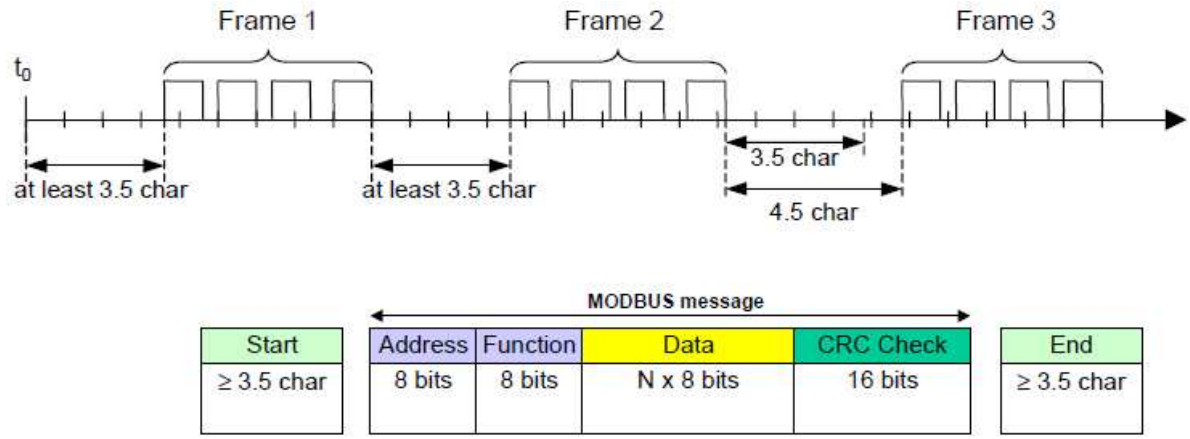
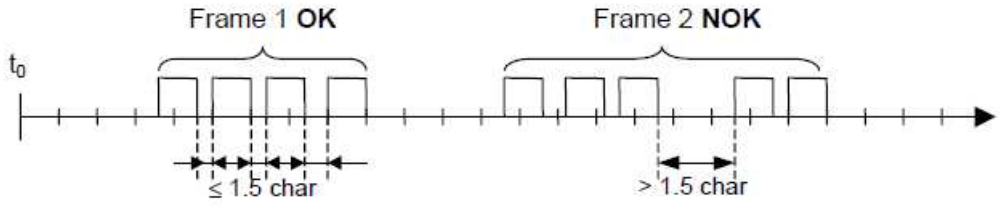


Figure 13: RTU Message Frame

The entire message frame must be transmitted as a continuous stream of characters.

If a silent interval of more than 1.5 character times occurs between two characters, the message frame is declared incomplete and should be discarded by the receiver.



Remark:
The implementation of RTU reception driver may imply the management of a lot of interruptions due to the $t_{1.5}$ and $t_{3.5}$ timers. With high communication baud rates, this leads to a heavy CPU load. Consequently these two timers must be strictly respected when the baud rate is equal or lower than 19200 Bps. For baud rates greater than 19200 Bps, fixed values for the 2 timers should be used: it is recommended to use a value of 750µs for the inter-character time-out ($t_{1.5}$) and a value of 1.750ms for inter-frame delay ($t_{3.5}$).

The following drawing provides a description of the RTU transmission mode state diagram. Both “master“ and “slave“ points of view are expressed in the same drawing:

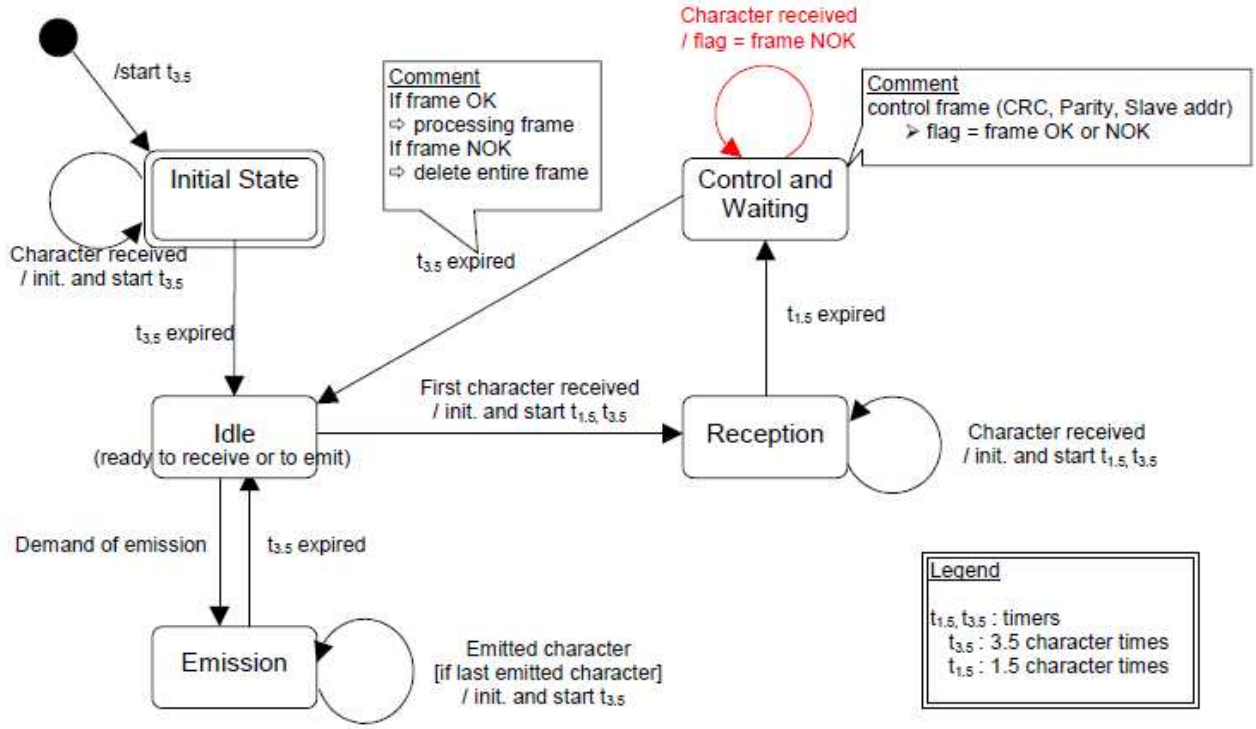


Figure 14: RTU transmission mode state diagram

Some explanations about the above state diagram:

- Transition from “Initial State“ to “Idle“ state needs $t_{3.5}$ time-out expiration: that insures inter-frame delay
- “Idle“ state is the normal state when neither emission nor reception is active.
- In RTU mode, the communication link is declared in “idle“ state when there is no transmission activity after a time interval equal to at least 3,5 characters.
- When the link is in idle state, each transmitted character detected on the link is identified as the **start of a frame**. The link goes to the “active“ state. Then, the **end of frame** is identified when no more character is transmitted on the link after the time interval t3,5.
- After detection of the end of frame, the CRC calculation and checking is completed. Afterwards the address field is analysed to determine if the frame is for the device. If not the frame is discarded. In order to reduce the reception processing time the address field can be analysed as soon as it is received without waiting the end of frame. In this case the CRC will be calculated and checked only if the frame is addressed to the slave (broadcast frame included).

Annexe 7 : Modules TDA08/TDA04

TDA08/TDA04

Protocole de communication

Extraits du manuel d'utilisation

SOMMAIRE :
(Note : plusieurs paragraphes qui ne sont pas utiles pour répondre aux questions du sujet ont été retirés)

1 Introduction

Ce document a le but de décrire les capacités de communication de tous les modules d'acquisition TDA qui utilisent le protocole MODBUS et il est surtout adressé aux techniciens, intégrateurs de systèmes et créateurs de logiciel.

Il est subdivisé en quatre parties :

- la première décrit la connexion physique à la ligne ;
- la seconde présente le protocole de communication, qui est un sous-ensemble du MODBUS RTU[1] ;
- la troisième partie décrit les différents types de données qui peuvent être échangées ;
- la quatrième reporte les performances typiques du système.

2 Connexion physique

2.1 Interface

Les modules TDA sont munis d'interface de communication sérielle optoisolée pour éviter l'apparition des problèmes dus aux potentiels de terre.

En position d'attente le module est en condition de réception et passe en transmission après avoir reçu et décodé un message correct qui lui est adressé.

Chaque module est muni d'un switch rotatif à 16 positions qui permet de programmer son adresse modbus. Les positions valables sont 15 (de 1 à 15, l'adresse ZERO est réservée par le MODBUS RTU pour les messages de broadcasting, mais elle n'est pas adoptée pour le TDA vu le manque de fiabilité implicite de ce type de communication).

Le tableau suivant illustre les programmations possibles :

Position switch rotative	Adresse du module	
	TDA08	TDA04
0	Non valable	Non valable
1	2 et 3	1
2	4 et 5	2
3	6 et 7	3
4	8 et 9	4
5	10 et 11	5
6	12 et 13	6
7	14 et 15	7
8	16 et 17	8
9	18 et 19	9
A	20 et 21	10
B	22 et 23	11
C	24 et 25	12
D	26 et 27	13
E	28 et 29	14
F	30 et 31	15

[1] Marque enregistrée par AEG Schneider Automation, Inc

N.B. : Chaque module TDA08 possède 2 adresses pour permettre au data-logger TMS01 d'enregistrer les huit possibles entrées de la sonde. Sur le TMS01, par exemple, il sera possible configurer deux dispositifs pour chaque module TDA08, le premier enregistrera les entrées IN1..IN4, le deuxième les entrées IN5..IN8.

Le baud rate de chaque module a comme programmation d'usine la valeur de 9600 baud. On peut le modifier par le modbus et la nouvelle programmation deviendra active au prochain cycle d'extinction-allumage du module.

3 Protocole de communication

Le protocole adopté par les TDA est un sous-ensemble du protocole largement utilisé MODBUS RTU. Ce choix garantit la facilité de connexion plusieurs PLC et à tous les programmes de supervision commerciaux.
Pour ceux qui veulent développer leur propre logiciel d'application toutes les suggestions et les informations sont disponibles.

Les fonctions du protocole MODBUS RTU implémentées dans les TDA sont :

fonction 3 - lecture de n mots
fonction 6 - écriture d'un mot

Ces fonctions permettent au programme de supervision de lire et modifier toute donne du module. La communication se base sur des messages envoyés par la station master à une station slave (TDA) et le contraire. La station slave qui reconnaît dans le message sa propre adresse, en analyse le contenu et, si elle le trouve formellement et sémantiquement correct, elle engendre un message de réponse pour le master.

Le procédé de communication implique cinq types de message :

du master au slave	du slave au master
fonction 3 : demande de lecture de n mots	fonction 3 : réponse contenant n mots lus
fonction 6 : demande d'écriture d'un mot	fonction 6 : confirmation de l'écriture d'un mot
	Réponse d'exception (en réponse aux deux fonctions, en cas d'anomalie)

Tout message contient quatre zones :

- adresse du slave : sont valables les valeurs comprises entre 1 et 31 (voir tableau a **2.1**); l'adresse 0 (zéro) est réservée par le MODBUS RTU pour les messages de broadcasting, mais il n'est pas adopté pour le TDA vu le manque de fiabilité de ce type de communication ;
- code fonction : contient 3 ou 6 selon la fonction spécifiée ;
- zone d'informations : contient les adresses ou la valeur des mots, selon la demande de la fonction utilisée ;
- mot de contrôle : contient un cyclic redundancy check (CRC) calculé selon les règles prévues pour le CRC16.

Les caractéristiques de la communication asynchrone sont : 8 bits, aucune parité, un bit d'arrêt.

3.1 Fonction 3 - lecture de n mots

Le nombre de mots à lire, doit être inférieur ou égal à quatre.

La demande a la structure suivante :

numéro du slave	3	adresse premier mot MSB	LSB	nombre de mots MSB	LSB	CRC LSB	MSB
byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7

La réponse normale (au contraire d'une réponse d'exception) a la structure suivante :

numéro du slave	3	nombre de bytes lus	valeur du premier mot MSB	LSB	mots suivants	CRC LSB	MSB
byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte	byte

3.2 Fonction 6 - écriture d'un mot

La demande a la structure suivante :

numéro du slave	6	Adresse premier mot MSB	LSB	Valeur à écrire MSB	LSB	CRC LSB	MSB
byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7

La réponse normale (au contraire de la réponse d'exception) est purement un écho du message de demande :

numéro du slave	6	Adresse premier mot MSB	LSB	Valeur à écrire MSB	LSB	CRC LSB	MSB
byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	Byte 6	byte 7

4 Echange des données

Cette section contient les informations concernant les données numériques et non numériques échangées avec les modules TDA et leurs formats et limites.

4.1 Certaines définitions

Toutes les données échangées sont constituées par un mot de 16 bits.

On distingue deux types de données : numériques et symboliques (ou no numériques). Les données numériques représentent la valeur d'une grandeur (par exemple la variable mesurée, etc…).

Les données symboliques représentent une valeur particulière à l'intérieur d'une gamme de choix (par exemple, Unité de mesure peut valoir "°C" ou "°F").

Les deux types sont codifiés avec des numéros entiers : on adopte des numéros entiers avec signe pour les données numériques et les numéros entiers sans signe pour les symboliques. Une donnée numérique doit être associée avec le numéro approprié de chiffres décimaux, de façon à représenter une grandeur avec les mêmes unités d'ingénierie adoptées dans le module TDA.

Les données numériques sont représentées avec une virgule fixe, et peuvent être entières ou avec un chiffre décimal.

4.2 Zones de mémoire

Pour les fonctions adoptées, toutes les données lisibles et que l'on peut écrire apparaissent comme des mots de 16 bits placés dans la mémoire du module.

Le plan de la mémoire a cinq zones :

Paramètres,
variables,
commandes, alarmes,
code d'identification de l'instrument.

Les paragraphes suivants examinent les caractéristiques de chaque zone.

Un appendice approprié énumère tous les détails de chaque zone, de façon à permettre la connexion à un système de supervision.

4.2.1 Zone des paramètres

Les données de configuration ainsi que les données opérationnelles se trouvent dans la zone des paramètres et sont physiquement dans une mémoire non volatile située à l'intérieur des TDA.

4.2.2 Zone des variables

Dans cette zone, on a regroupé les variables principales du TDA qui sont fréquemment calculées et mises à jour.

On énumère ici les données disponibles :

- v valeur mesurée de la sonde 1,
- v valeur mesurée de la sonde 2,
- v valeur mesurée de la sonde 3,
- v valeur mesurée de la sonde 4,
- v valeur mesurée de la sonde 5,
- v valeur mesurée de la sonde 6,
- v valeur mesurée de la sonde 7,
- v valeur mesurée de la sonde 8,
- v état des entrées digitales,
- v état de la sortie,
- v état des alarmes,
- v état du TDA,

Les conditions d'anomalie des variables de procédé (sonde 1...sonde 8) sont reportées comme des valeurs spéciales de la mesure :

condition d'anomalie	valeur rendue
Underrange ou court-circuit	-10000
Overflow ou sonde ouverte	10000
Variable non disponible	10003

A.2 Appendice - Tableau de la zone des variables

n.	*adresse (hex)*	*nom variable*	*type donnée*	*Étendue de mesure*	*unité*	*chiffres décimaux*	*r/w*
1	0200	Valeur entrée IN1	N	-999 … 9999	(*)	Var 0240	r
2	0201	Valeur entrée IN2	N	-999 … 9999	(*)	Var 0241	r
3	0202	Valeur entrée IN3	N	-999 … 9999	(*)	Var 0242	r
4	0203	Valeur entrée IN4	N	-999 … 9999	(*)	Var 0243	r
(#)5	0204	Valeur entrée IN5	N	-999 … 9999	(* *)	Var 0244	r
(#)6	0205	Valeur entrée IN6	N	-999 … 9999	(* *)	Var 0245	r
(#)7	0206	Valeur entrée IN7	N	-999 … 9999	(* *)	Var 0246	r
(#)8	0207	Valeur entrée IN8	N	-999 … 9999	(* *)	Var 0247	r
9	021F	Lit l'état de la sortie alarm OUT	S	0: OFF 1: ON			r
10	0220	Lit l'état de l'entrée DI01	S	0: ouvert 1: fermé			r
11	0221	Lit l'état de l'entrée DI02	S	0: ouvert 1: fermé			r
12	0222	Lit l'état de l'entrée DI03	S	0: ouvert 1: fermé			r
13	0223	Lit l'état de l'entrée DI04	S	0: ouvert 1: fermé			r
(#)14	0224	Lit l'état de l'entrée DI05 (IN1 quand Endi=YES)	S	0: ouvert 1: fermé			r
(#)15	0225	Lit l'état de l'entrée DI06 (IN2 quand Endi=YES)	S	0: ouvert 1: fermé			r
(#)16	0226	Lit l'état de l'entrée DI07 (IN3 quand Endi=YES)	S	0: ouvert 1: fermé			r
(#)17	0227	Lit l'état de l'entrée DI08 (IN4 quand Endi=YES)	S	0: ouvert 1: fermé			r
(#)18	0228	Lit l'état de l'entrée DI09 (IN5 quand Endi=YES)	S	0: ouvert 1: fermé			r
(#)19	0229	Lit l'état de l'entrée DI10 (IN6 quand Endi=YES)	S	0: ouvert 1: fermé			r
(#)20	022A	Lit l'état de l'entrée DI11 (IN7 quand Endi=YES)	S	0: ouvert 1: fermé			r