

# JUMO cTRON 04/08/16

Régulateur compact  
avec minuterie et fonction de rampe



702071



702072



702074

## B 70.2070.2.0

Description de l'interface  
Modbus



# Sommaire

---

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Avant-propos .....	5
1.2	Conventions typographiques .....	5
<b>2</b>	<b>Description du protocole</b>	<b>7</b>
2.1	Principe maître-esclave .....	7
2.2	Mode de transmission (RTU) .....	7
2.3	Adresse-appareil .....	8
2.4	Déroulement temporel de la transmission .....	8
2.5	Structure des blocs de données .....	11
2.6	Code des fonctions .....	12
2.6.1	Lecture de n mots .....	12
2.6.2	Ecriture d'un mot .....	13
2.6.3	Ecriture de n mots .....	14
2.7	Format de transmission (valeurs entières, flottantes et texte) .....	15
2.8	Somme de contrôle (CRC16) .....	17
2.9	Traitement des erreurs .....	18
<b>3</b>	<b>Interface RS485</b>	<b>19</b>
3.1	Schéma de raccordement .....	19
3.2	Configuration .....	20
<b>4</b>	<b>Adresses Modbus</b>	<b>21</b>
4.1	Données de process .....	21
4.2	Consignes .....	23
4.3	Paramètres du régulateur .....	23
4.4	Configuration .....	23
4.5	Commandes .....	24
4.6	Mémoire RAM .....	25

---



## 1.1 Avant-propos

Cette notice s'adresse aux constructeurs avec formation spécialisée et possédant des connaissances en PC.

Lisez cette notice avant de mettre en service l'interface.

Conservez cette notice dans un endroit accessible à tout moment à tous les utilisateurs. Aidez-nous à améliorer cette notice en nous faisant part de vos suggestions.

Tous les réglages nécessaires sont décrits dans cette notice. Si vous procédez à des manipulations non décrites dans cette notice ou expressément interdites, vous compromettez votre droit à la garantie. En cas de problèmes, veuillez prendre contact avec nos services.

## 1.2 Conventions typographiques

### Signes d'avertissement :

#### Attention



Ce symbole est utilisé lorsque la non-observation ou l'observation imprécise des instructions peut **endommager les appareils ou les données** !

### Symboles indiquant une remarque :

#### Remarque



Ce symbole est utilisé pour attirer votre attention sur un **point particulier**.

#### Renvoi



Ce symbole renvoie à des **informations complémentaires** dans d'autres notices, chapitres ou sections.

### Types de représentation:

#### Nombre hexadécimal

0x0010

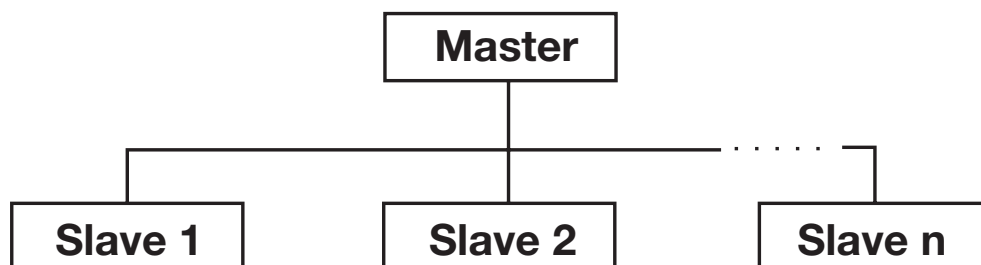
Un nombre hexadécimal est précédé de "0x" (ici 16 décimales).

# 1 Introduction

---

### 2.1 Principe maître-esclave

La communication entre un appareil maître (par ex. un PC) et un appareil esclave (par ex. système de mesure et de régulation) avec le protocole ModBus se déroule selon le principe maître/esclave sous la forme demande de données/instruction-réponse.



Le maître contrôle l'échange de données, les esclaves ne donnent que des réponses. Ils sont identifiés par leur adresse d'appareil.

### 2.2 Mode de transmission (RTU)

Le mode de transmission est le mode RTU (Remote Terminal Unit). La transmission des données s'effectue sous forme binaire (hexadécimale) sur 8 bits. Le bit de poids faible (LSB = least significant bit) est transmis en premier. Le code ASCII n'est pas pris en considération.

#### Format des données

Le format des données décrit la structure d'un symbole transmis. Les différents formats de données possibles sont les suivants :

Mot de données	Bit de parité	Bit de stop 1/2 bit(s)	Nombre de bits
8 Bit	—	1	9
8 Bit	paire (even)	1	10
8 Bit	impaire (odd)	1	10
8 Bit	—	2	10

## 2 Description du protocole

---

### 2.3 Adresse-appareil

L'adresse appareil de l'esclave est réglable entre 0 et 254.  
L'adresse appareil 0 est réservée.



L'interface RS485 permet d'adresser au maximum 31 esclaves.

Il existe deux possibilités d'échange de données :

#### **Query (consultation)**

Demande de données / instruction du maître à un esclave au travers d'une adresse d'appareil particulière.

L'esclave adressé répond.

#### **Broadcast (diffusion)**

Instruction du maître à tous les esclaves à l'aide de l'adresse d'appareil 0 (par ex. pour la transmission d'une valeur définie à tous les esclaves).

Les esclaves connectés ne répondent pas. La réception correcte de la valeur par les esclaves devra être contrôlée par une lecture ultérieure de la consigne.

Une demande de données avec l'adresse d'appareil 0 n'est pas logique.

### 2.4 Déroulement temporel de la transmission

Le début et la fin d'un bloc de données sont caractérisés par des pauses de transmission. Entre deux caractères consécutifs, il doit s'écouler au maximum trois fois le temps de transfert d'un caractère.

Le temps de transfert d'un caractère dépend du débit en baud et du format de données utilisé (nombre de bits de stop et parité).

Pour le format de données 8 bits, sans bit de parité et avec un bit de stop, le temps de transfert d'un caractère est égal à :

**Temps de transfert d'un caractère [ms] = 1000 \* 9 bits / débit en baud**

Pour les autres formats de données :

**Temps de transfert d'un caractère [ms] =  
1000 \* (8 bits + bit de parité + bit(s) de stop) / débit en baud**



## 2 Description du protocole

### Déroulement

<b>Demande de données du maître</b> Temps de transfert = $n \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$
Identificateur de fin de demande de données $3 \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$
Traitement de la demande de données par l'esclave ( $\leq 250 \text{ ms}$ )
<b>Réponse de l'esclave</b> Temps de transfert = $n \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$
Identificateur de fin de réponse $3 \text{ caractères} * 1000 * x \text{ bits} / \text{débit en baud}$

### Exemple

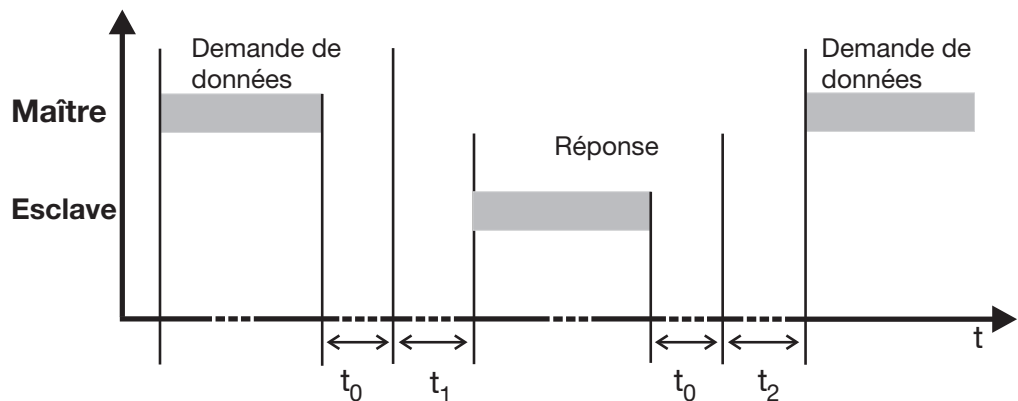
Identificateurs de fin de demande de données et de réponse pour le format 10/9 Bits

Temps d'attente =  $3 \text{ caractères} * 1000 * 10 \text{ bits} / \text{débit en baud}$

Débit [bauds]	Format de données [bit]	Temps d'attente [ms] (3 caractères)
38400	10	0.79
	9	0.71
19200	10	1.57
	9	1.41
9600	10	3.13
	9	2.82

## 2 Description du protocole

**Chronogramme** Une demande de données se déroule selon le chronogramme suivant :



$t_0$  Identificateur de fin = 3 caractères.

La durée dépend du débit en baud.

$t_1$  Cette durée dépend du traitement interne.

La durée maximale de traitement est de 250 ms.



Dans l'appareil, sous le point du menu "Interface", il faut régler le temps de réponse minimal. Il faut laisser s'écouler cette durée avant d'envoyer une réponse (0 à 500 ms). Si on règle une valeur faible, le temps de réponse peut être supérieur à la valeur réglée (le traitement interne dure plus longtemps), l'appareil répond dès que le traitement interne est terminé. Si on règle la valeur sur 0 ms, l'appareil répondra le plus rapidement possible.

Avec une interface RS485, le maître a besoin du temps minimal de réponse (réglable) pour commuter l'interface d'émission en réception.

$t_2$  L'esclave a besoin de ce temps pour recommuter de l'émission en réception. Le maître laisse s'écouler ce temps avant de poser une nouvelle demande de données. Ce temps doit toujours être respecté, même si la nouvelle demande de données est envoyée à un autre appareil.

Interface RS485 :  $t_2 = 10\text{ms}$

Le maître ne peut émettre une demande de données, ni à l'intérieur de  $t_1$  et  $t_2$ , ni pendant le temps de réponse. Les demandes pendant  $t_1$  et  $t_2$  sont ignorées par l'esclave. Les demandes pendant le temps de réponse ont pour conséquence que toutes les données qui se trouvent à ce moment sur le bus deviennent invalides.

## 2 Description du protocole

---

### 2.5 Structure des blocs de données

Tous les blocs de données ont la même structure :

#### Structure des données

Adresse de l'esclave	Code de la fonction	Données	Somme de contrôle CRC16
1 octet	1 octet	x octet	2 octets

Chaque bloc de données dispose de 4 champs :

<b>Adresse de l'esclave</b>	Adresse d'appareil d'un certain esclave
<b>Code de la fonction</b>	Choix de la fonction (lecture, écriture de mots)
<b>Données</b>	Contient les informations : <ul style="list-style-type: none"><li>- adresse des mots</li><li>- nombre de mots</li><li>- valeur des mots</li></ul>
<b>Somme de contrôle</b>	Détection des erreurs de transmission

## 2 Description du protocole

### 2.6 Code des fonctions

Les fonctions suivantes décrites sont disponibles pour la lecture de valeurs de mesure, de données de process et des appareils ainsi que pour l'écriture de données définies.

#### Aperçu des fonctions

Numéro	Fonction	Limitation
0x03 ou 0x04	Lecture de n mots	max. 32 mots (64 octets)
0x06	Ecriture d'un mot	max. 1 mot (2 octets)
0x10	Ecriture de n mots	max. 32 mots (64 octets)



Lorsque l'appareil ne réagit pas à ces fonctions ou émet un code d'erreur, voir chapitre 2.9 Traitement des erreurs, page 18.

#### 2.6.1 Lecture de n mots

Avec cette fonction, n mots ( $n \leq 32$ ) peuvent être lus à partir d'une adresse définie

#### Demande de données

Adresse de l'esclave	Fonction x03 ou 0x04	Adresse 1er mot	Nombre de mots (max. 32)	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

#### Réponse

Adresse de l'esclave	Fonction 0x03 ou 0x04	Nombre d'octets lus	Valeur mot(s)	Somme de contrôle CRC16
1 octet	1 octet	1 octet	x octet(s)	2 octets

#### Exemple

Lecture des consignes W1 et W2 (respectivement 2 mots)

Adresse du premier mot = 0x3100 (consigne W1)

Demande de données :

01	03	3100	0004	4AF5
----	----	------	------	------

Réponse (valeurs en format Modbus flottant):

01	03	08	0000	41C8	0000	4120	4A9E
			consigne W1 (25.0)		consigne W2 (10.0)		

## 2 Description du protocole

### 2.6.2 Ecriture d'un mot

Pour cette fonction, les blocs de données de l'ordre sont identiques aux blocs de données de la réponse.

#### Ordre

Adresse de l'esclave	Fonction 0x06	Adresse mot	Valeur mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octet

#### Réponse

Adresse de l'esclave	Fonction 0x06	Adresse mot	Valeur mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

#### Exemple

Ecriture de la valeur limite AL seuil d'alarme 1 = 275.0  
(valeur = 0x80004389 en format Modbus flottant)

Adresse mot = 0x0057

Ordre : écriture de la première partie de la valeur

01	06	0057	8000	59DA
----	----	------	------	------

Réponse (idem ordre) :

01	06	0057	8000	59DA
----	----	------	------	------

Ordre : écriture de la seconde partie de la valeur (prochaine adresse mot)

01	06	0058	4389	F88F
----	----	------	------	------

Réponse (idem ordre) :

01	06	0058	4389	F88F
----	----	------	------	------

## 2 Description du protocole

---

### 2.6.3 Ecriture de n mots

Avec cette fonction, n mots ( $n \leq 32$ ) peuvent être lus à partir d'une adresse définie.

#### Ordre

Adresse de l'esclave	Fonction 0x10	Adresse 1er mot	Nombre de mot (max. 32)	Nombre d'octets	Valeur(s) mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	1 octet	x octet(s)	2 octets

#### Réponse

Adresse de l'esclave	Fonction 0x10	Adresse 1er mot	Nombre de mot	Somme de contrôle CRC16
1 octet	1 octet	2 octets	2 octets	2 octets

#### Exemple

Ecriture des consignes W1 et W2

Adresse mot = 0x3100 (consigne W1)

Ordre :

01	10	3100	0004	08	0000	41C8	0000	4120	2A42
					Consigne W1 (25.0)		Consigne W2 (10.0)		

Réponse:

01	10	3100	0004	CF36
----	----	------	------	------

## 2 Description du protocole

### 2.7 Format de transmission (valeurs entières, flottantes et texte)

**Valeurs entières** Avec le protocole Modbus, les valeurs entières sont transmises sous la forme suivante : d'abord l'octet de poids fort, ensuite l'octet de poids faible.

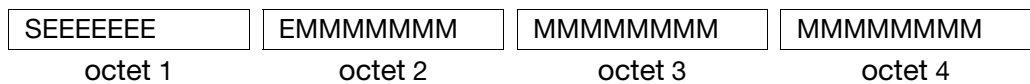
**Exemple** Consultation de la valeur entière à l'adresse 0x0021 lorsque le contenu à cette adresse est "4" (valeur du mot 0x0004).

Demande : 01 03 0021 0001 (+ 2 octets CRC16)

Réponse : 01 03 02 **0004** (+ 2 octets CRC16)

**Valeurs flottantes** Le protocole Modbus traite les valeurs flottantes conformément au format standard IEEE 754 (32 bits) ; toutefois il y a une différence : les octets 1 et 2 sont échangés avec les octets 3 et 4.

**Format des valeurs flottantes simples (32 bits) suivant le standard IEEE 754**

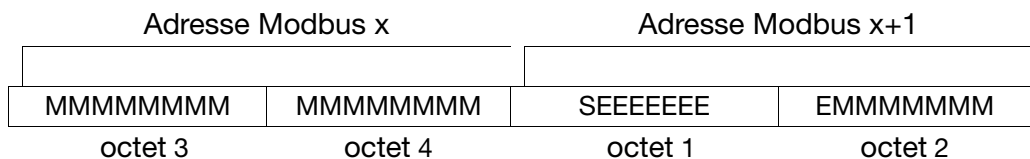


S - Bit de signe

E - Exposant (complément à 2)

M - Mantisse normalisée sur 23 bits

**Format flottant Modbus**



**Exemple** Consultation de la valeur flottante à l'adresse 0x0035 lorsque le contenu à cette adresse est 550.0 (0x44098000 au format IEEE 754).

Demande : 01 03 0035 0002 (+ 2 octets CRC16)

Réponse : 01 03 04 **8000 4409** (+ 2 octets CRC16)

Après le transfert depuis l'appareil, il faut échanger les octets de cette valeur.



De nombreux compilateurs (par ex. Microsoft Visual C++) manipulent les valeurs flottantes dans l'ordre suivant :

**Valeur flottante**



Déterminez le mode de stockage des valeurs flottantes dans votre application. Le cas échéant, après la consultation de l'enregistreur sans papier, il faudra échanger les octets dans votre programme d'interface.

## 2 Description du protocole

---

### Chaînes de caractères (textes)

Les chaînes de caractères (textes) sont transmises en format ASCII.



Le dernier caractère (indicateur de fin) doit toujours être un "\0" (code ASCII 0x00). Les caractères qui suivent n'ont aucune importance.

Le nombre de caractères max. dans le type de données est indiqué dans les tableaux d'adresse, par ex. "TEXT24" (24 caractères). Si l'on utilise aucun indicateur de fin, il ne reste dans ce cas de figure que 23 caractères lisibles pour le texte.

Si l'on utilise aucun indicateur de fin, il faut, pour l'écriture que le nombre de caractères max. indiqué dans le type de données (par ex. TEXT8 = 8 caractères) soit utilisé. On évite ainsi que des caractères encore en mémoire soient annexés au texte.

Comme la transmission des textes a lieu également mot par mot (16 bits), il faut envoyer un 0x00 supplémentaire si le nombre de caractères est impair (y compris le caractère "\0").

### Exemple pour type de données TEXT4

Lecture du texte (ici : "AbC ") sous l'adresse 0x0067 (4 caractères max. peuvent être mémorisés)

Code ASCII pour "AbC " (avec 1 espace à la fin) :  
**0x41, 0x62, 0x43, 0x20**

Demande : 01 03 0067 0002 (+ 2 octets CRC16)

Adresse esclave = 01

Fonction = 03, c.-à-d. lecture de n mots

Adresse = 0067

Nombre de mots à lire = 0002, 4 caractères max.

Réponse : 01 03 04 **41 62 43 20** (+ 2 octets CRC16)

Adresse esclave = 01

Fonction = 03, c.-à-d. lecture de n mots

Nombre d'octets lus = 04

Variante :

Code ASCII pour "Ab" (sans espace à la fin) :

**0x41, 0x62, 0x00**

ASCII 0x00 ("\0") signifie que la chaîne de caractères s'arrête ici.

Lors de la transmission un **0x00** supplémentaire est annexé, pour obtenir un nombre pair de caractères.

Réponse dans ce cas : 01 03 04 **41 62 00 00** (+ 2 octets CRC16)



## 2 Description du protocole

### 2.8 Somme de contrôle (CRC16)

La somme de contrôle (CRC16) permet de détecter les erreurs de transmission. Si une erreur est détectée lors de l'analyse, l'appareil correspondant ne répond pas.

#### Mode de calcul

CRC = 0xFFFF	
CRC = CRC XOR ByteOfMessage	
For (1 à 8)	
CRC = SHR(CRC)	
if (drapeau report à droite = 1)	
then	else
CRC = CRC XOR 0xA001	
while (tous les octets du message ne sont pas traités);	



L'octet faible de la somme de contrôle est transféré le premier, suivi de l'octet fort.

#### Exemple

Demande de données : lecture de 2 mots à partir de l'adresse 0x00CE  
(CRC16 = 0x92A5)

07	03	00	CE	00	02	A5	92
							CRC16

Réponse : (CRC16 = 0xF5AD)

07	03	04	00	00	41	C8	AD	F5
				mot 1	mot 2		CRC16	

## 2 Description du protocole

---

### 2.9 Traitement des erreurs

**Code d'erreur** Codes d'erreur possibles :

- 1 fonction invalide
- 2 Adresse de paramètres invalide ou nombre de mots ou de bits à lire ou à écrire trop élevé
- 8 Paramètre protégé en écriture

**Réponse en cas d'erreur**

Adresse esclave	Fonction XX OR 80h	Code d'erreur	Somme de contrôle CRC16
1 octet	1 octet	1 octet	2 octets

Le code de la fonction est associé à 0x80 à l'aide d'une fonction OU (OR), c'est-à-dire que le bit de poids fort (MSB = Most Significant Bit) est mis à 1:

**Exemple**

Demande de données :

01	03	40	00	00	04	CRC16
----	----	----	----	----	----	-------

Réponse (avec code d'erreur 2) :

01	83	02	CRC16
----	----	----	-------

**Cas particuliers**

Dans les cas suivants, l'esclave ne répond pas :

- débit en baud et/ou format de données ne concordent pas pour le maître et pour l'esclave
- L'adresse de l'appareil utilisée ne concorde pas avec l'adresse de l'esclave
- La somme de contrôle (CRC16) est incorrecte.
- L'ordre du maître est incomplet ou contradictoire
- Le nombre de mots à lire est égal à 0

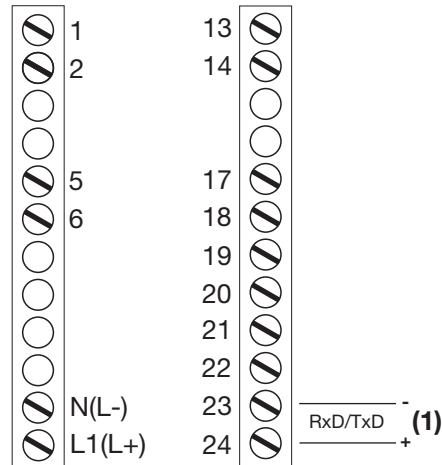
Dans ces cas, la demande de données devrait après écoulement du temps du timeout (2s) être renvoyée.

## 3.1 Schéma de raccordement

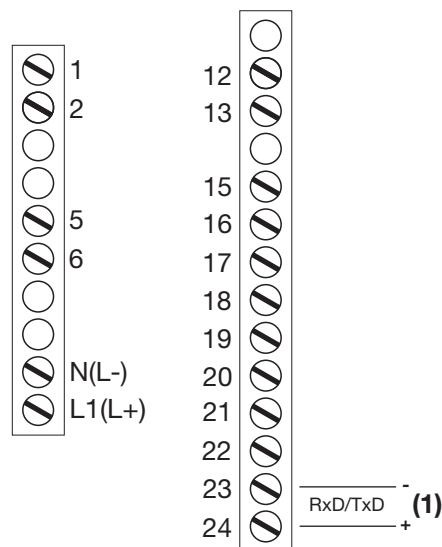


Les appareils de cette série de régulateurs peuvent être commandés avec une interface RS485 en option. Vous trouverez les références de commande dans la fiche technique 70.2070 ou dans la notice de mise en service B 70.2070.0 (identification du type).

### Schéma de raccordement Type 702071



### Schéma de raccordement Type 702072 et type 702074



(1) Interface RS485



Le blindage du câble de l'interface doit être mis unilatéralement à la terre dans l'armoire de commande.

## 3 Interface RS485

### 3.2 Configuration

Les différents réglages de l'interface Modbus qui sont effectués au niveau de configuration (*Conf* → *IntF*) et/ou dans le logiciel Setup sont décrits dans le tableau ci-dessous.



Vous trouverez des informations complémentaires pour la configuration dans la notice de mise en service B 70.2070.0.

	<b>Symbole</b>	<b>Valeur / Sélection</b>	<b>Description</b>
<b>Débit en baud</b>	<i>bdr</i>	<b>0</b>	<b>9600 Baud</b>
		1	19200 Baud
		2	38400 Baud
<b>Format des données</b>	<i>dft</i>	<b>0</b>	<b>8 bits de données, 1 bit stop, pas de parité</b>
		1	8 bits de données, 1 bit stop, parité impaire
		2	8 bits de données, 1 bit stop, parité paire
		3	8 bits de données, 2 bits stop, pas de parité
<b>Adresse app.</b>	<i>Adr</i>	0 à 1 à 255	Adresse dans le réseau de données Les adresses 0 et 255 sont prévues à des fins définies et ne devraient être utilisées ici.
<b>Temps de réponse min.</b>	(Setup)	<b>0</b> à 500ms	Laps de temps, entre le moment de la demande d'un appareil dans un réseau de données et la réponse du régulateur (réglable uniquement via le logiciel Setup).

Les réglages d'usine sont en **gras**.



L'interface RS485 est inactive si communication via l'interface Setup.

## 4 Adresses Modbus

### Type de données, type d'accès

Vous trouverez dans le tableau ci-dessous toutes les valeurs de process (variables) avec leurs adresses, le type de données ainsi que le mode d'accès.

Signification :

<b>R/O</b>	Lecture seule
<b>W/O</b>	Ecriture seule
<b>R/W</b>	Ecriture et lecture
<b>INT</b>	Integer (8 ou 16 Bit)
<b>Bit x</b>	Bit No x (Bit 0 est le bit le plus faible)
<b>LONG</b>	Long Integer (4 octets)
<b>FLOAT</b>	Valeur flottante (4 octets) suivant IEEE 754
<b>TEXT4</b>	Texte à 4 caractères



Les opérations d'écriture sur les paramètres R/W provoquent une sauvegarde dans l'EEPROM. Ces modules mémoires n'ont qu'un nombre limité de cycles d'écriture (env. 10 000), c'est pourquoi cette fonction peut être déconnectée en cas de programmation fréquente. Les valeurs de paramètres sont alors seulement enregistrées dans la mémoire volatile (RAM) et sont perdues en cas de panne de secteur.

⇒ *Setup/Uniquement Setup/Paramètre non documenté/Paramètre Bit/Activer paramètre 2*

### 4.1 Données de process

Adresses	Type de données / numéro de bit	Accès	Désignation du signal
0x001F	INT	R/O	Valeurs binaires internes
	Bit 12		Valeur binaire L1 (= 0x1000)
	Bit 13		Valeur binaire L2 (= 0x2000)
0x0020	INT	R/O	Etat du régulateur
	Bit 12		Mode manuel actif (= 0x1000)
	Bit 15		Auto-optimisation active (= 0x8000)
0x0021	INT	R/O	Sorties binaires 1 à 4 (états de commutation 0 = OFF / 1 = ON)
	Bit 0		Sortie K1: relais (= 0x0001)
	Bit 1		Sortie K2: relais (= 0x0002)
	Bit 2		Sortie K3: logique (= 0x0004)
	Bit 3		Sortie K4: relais (= 0x0008)

## 4 Adresses Modbus

Adresses	Type de données / numéro de bit	Accès	Désignation du signal
0x0023	INT	R/O	Entrées binaires 1 et 2 (états de commutation 0 = ouverte / 1 = fermée)
	Bit 0		Entrée 1 (= 0x0001)
	Bit 1		Entrée 2 (= 0x0002)
0x0024	INT	R/O	Seuils d'alarme 1 à 2
	Bit 0		Seuil d'alarme 1 (= 0x0001)
	Bit 1		Seuil d'alarme 2 (= 0x0002)
0x0025	INT	R/W	Commande Sorties binaires (séparément)
	Bit 0 + Bit 8		Sortie K1 (= 0x0101)
	Bit 1 + Bit 9		Sortie K2 (= 0x0202)
	Bit 2 + Bit 10		Sortie K3 (= 0x0404)
	Bit 3 + Bit 11		Sortie K4 (= 0x0808)
0x0026	FLOAT	R/O	Entrée analogique [mV]
0x0028	FLOAT	R/O	Pt100 Interne [Ohm]
0x002A	INT	R/O	Cadence de scrutation
0x002B	FLOAT	R/O	Entrée analogique [valeur d'affichage]
0x002D	FLOAT	R/O	Valeur analogique interne 1
0x002F	FLOAT	R/O	Valeur analogique interne 2
0x0031	FLOAT	R/O	Régulateur Valeur fin de rampe
0x0033	FLOAT	R/O	Régulateur Valeur réelle FILTEREE
0x0035	FLOAT	R/O	Régulateur Valeur réelle NON FILTEREE
0x0037	FLOAT	R/W	Régulateur Consigne
0x0039	FLOAT	R/O	Régulateur Affichage du taux de modulation
0x003B	FLOAT	R/O	Régulateur Taux de modulation CHAUFFER
0x003D	FLOAT	R/O	Régulateur Taux de modulation REFROIDIR
0x003F	FLOAT	R/O	Régulateur Différence
0x0041	FLOAT	R/O	Régulateur Ecart de régulation
0x0043	INT	R/O	Régulateur Position du point de contact CHAUFFER
0x0044	INT	R/O	Régulateur Position du point de contact REFROIDIR
0x0046	INT	R/O	Taux de modulation manuel
0x0047	LONG	R/O	Temps d'exécution de la minuterie
0x0049	LONG	R/O	Temps restant de la minuterie
0x004B	INT	R/O	Etat de la minuterie
	Bit 1		Minuterie arrêtée (= 0x0002)
	Bit 5		Minuterie fonctionne (= 0x0020)
	Bit 6		Fin minuterie (= 0x0040)
	Bit 15		Signal de la minuterie (= 0x8000)

### 4.2 Consignes

Adresses	Type de données / numéro de bit	Accès	Désignation du signal
0x3100	FLOAT	R/W	Consigne W1
0x3102	FLOAT	R/W	Consigne W2

### 4.3 Paramètres du régulateur

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x3000	FLOAT	R/W	Paramètre du régulateur XP1
0x3002	FLOAT	R/W	Paramètre du régulateur XP2
0x3004	FLOAT	R/W	Paramètre du régulateur TV
0x3006	FLOAT	R/W	Paramètre du régulateur TN
0x300C	FLOAT	R/W	Paramètre du régulateur CY1
0x300E	FLOAT	R/W	Paramètre du régulateur CY2
0x3010	FLOAT	R/W	Paramètre du régulateur XSH
0x3012	FLOAT	R/W	Paramètre du régulateur XD1
0x3014	FLOAT	R/W	Paramètre du régulateur XD2
0x3016	INT	R/W	Paramètre du régulateur TT
0x3017	INT	R/W	Paramètre du régulateur Y0
0x3018	INT	R/W	Paramètre du régulateur Y1
0x3019	INT	R/W	Paramètre du régulateur Y2

### 4.4 Configuration

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x0053	FLOAT	R/W	Fonction rampe Pente
0x0055	FLOAT	R/W	Constante de temps du filtre
0x0057	FLOAT	R/W	Seuil d'alarme 1 Valeur limite AL
0x0059	FLOAT	R/W	Seuil d'alarme 1 Différentiel de coupure
0x005D	FLOAT	R/W	Seuil d'alarme 2 Valeur limite AL
0x005F	FLOAT	R/W	Seuil d'alarme 2 Différentiel de coupure
0x0063	LONG	R/W	Valeur de la minuterie
0x0065	LONG	R/W	Valeur limite Intervention
0x0067	TEXT4	R/W	Texte d'alarme
0x0069	LONG	R/W	Compteur d'intervention

## 4 Adresses Modbus

---

### 4.5 Commandes

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x004D	INT	W/O	Fonctions binaires REGULATEUR
	Bit 0		Démarrage de l'auto-optimisation (= 0x0001)
	Bit 1		Annulation de l'auto-optimisation (= 0x0002)
	Bit 2		Mode manuel (= 0x0004)
	Bit 3		Mode automatique (= 0x0008)
	Bit 4		Régulateur OFF (= 0x0010)
	Bit 5		Verrouillage du mode manuel (= 0x0020)
	Bit 6		Stop Rampe (= 0x0040)
	Bit 7		Annulation Rampe (= 0x0080)
	Bit 8		Redémarrage Rampe (= 0x0100)
	Bit 9		Démarrage de la minuterie (= 0x0200)
	Bit 10		Annulation de la minuterie (= 0x0400)
	Bit 11		Arrêt de la minuterie (= 0x0800)
0x004E	INT	W/O	Fonctions binaires COMMANDE
	Bit 0		Verrouillage du clavier (= 0x0001)
	Bit 1		Verrouillage des niveaux de configuration et de paramétrage (= 0x0002)
	Bit 3		Affichage OFF (= 0x0008)
	Bit 5		Affichage textes (= 0x0020)
0x004F	INT	W/O	Fonctions binaires MINUTERIE
	Bit 9		Démarrer minuterie (= 0x0200)
	Bit 10		Annuler minuterie (= 0x0400)
	Bit 11		Arrêter minuterie (= 0x0800)
0x0050	INT	R/W	Commutation de la consigne
	Bit 0		Consigne 1 (= 0x0001)
	Bit 1		Consigne 2 (= 0x0002)



### 4.6 Mémoire RAM

Adresses	Type de données / numéro bit	Accès	Désignation du signal
0x3200	FLOAT	W/O	Consigne du régulateur (lecture/écriture)
0x3202	FLOAT	W/O	Valeur réelle du régulateur (lecture/écriture)
0x3204	FLOAT	W/O	Valeur analogique interne 1 (lecture/écriture)
0x3206	FLOAT	W/O	Valeur analogique interne 2 (lecture/écriture)
0x3208	INT	W/O	Valeurs binaires internes (lecture/écriture)
	Bit 0 + Bit 7		Valeur binaire L1 (= 0x0081)
	Bit 1 + Bit 7		Valeur binaire L2 (= 0x0082)



L'accès direct à la mémoire RAM de l'appareil est possible via le Modbus, afin d'écrire la valeur de consigne du régulateur (0x3200), la valeur réelle du régulateur (0x3202) et les valeurs analogiques internes (0x3204, 0x3206) ainsi que les valeurs binaires internes (0x3208).

Une plage comprise entre 1999 et +9999 est disponible pour l'écriture de la valeur du régulateur, la valeur réelle du régulateur et les valeurs analogiques internes. Est alors utilisée la valeur écrite dans l'appareil au lieu de la valeur originale.

Si la valeur originale doit à nouveau être utilisée, il faut écrire la valeur 200001 via le Modbus dans les emplacements mémoires respectifs.

# 4 Adresses Modbus

---





### **JUMO GmbH & Co. KG**

Adresse :

Moritz-Juchheim-Straße 1  
36039 Fulda, Allemagne

Adresse de livraison :

Mackenrodtstraße 14  
36039 Fulda, Allemagne

Adresse postale :

36035 Fulda, Allemagne

Téléphone : +49 661 6003-0

Télécopieur : +49 661 6003-607

E-Mail : [mail@jumo.net](mailto:mail@jumo.net)

Internet : [www.jumo.net](http://www.jumo.net)

### **JUMO Régulation SAS**

Actipôle Borny

7 Rue des Drapiers

B.P. 45200

57075 Metz - Cedex 3, France

Téléphone : +33 3 87 37 53 00

Télécopieur : +33 3 87 37 89 00

E-Mail : [info@jumo.fr](mailto:info@jumo.fr)

Internet : [www.jumo.fr](http://www.jumo.fr)

Service de soutien à la vente :

**0892 700 733** (0,337 Euro/min)

### **JUMO Automation**

**S.P.R.L. / P.G.M.B.H. / B.V.B.A.**

Industriestraße 18

4700 Eupen, Belgique

Téléphone : +32 87 59 53 00

Télécopieur : +32 87 74 02 03

E-Mail : [info@jumo.be](mailto:info@jumo.be)

Internet : [www.jumo.be](http://www.jumo.be)

### **JUMO Mess- und Regeltechnik AG**

Laubisrütistrasse 70

8712 Stäfa, Suisse

Téléphone : +41 44 928 24 44

Télécopieur : +41 44 928 24 48

E-Mail : [info@jumo.ch](mailto:info@jumo.ch)

Internet : [www.jumo.ch](http://www.jumo.ch)