

Important

Tous les renseignements et les spécifications techniques contenus dans le présent document ont été vérifiés attentivement et compilés par l'auteur. Cependant, nous ne pouvons pas exclure la présence d'erreurs. Il est toujours intéressant pour **Smith Meter GmbH** d'être notifié de toute erreur constatée.

TABLE DES MATIERES

1.	INTRODUCTION.....	5
1.1.	GENERALITÉS	5
1.2.	Références	5
1.3.	Abréviations	5
1.4.	Définitions	5
2.	DETAILS DU PROTOCOLE.....	7
2.1.	ModBus sur port série	7
2.1.1.	Port série 1	7
2.1.2.	Port série 2	7
2.1.3.	Détails de l'interface matérielle.....	8
2.1.4.	Délais à respecter pour RS-485	8
2.2.	ModBus sur TCP/IP	8
3.	MAPPAGE DE REGISTRES	9
3.1.	Informations de version.....	9
3.2.	Registres de résultats/de sortie.....	10
3.3.	Accumulateurs haute résolution	13
3.4.	Résultats de calcul DECA/GERG (MPU)	14
3.5.	Registres d'entrée	15
4.	UTILISATION DU REGISTRE	16
4.1.	État de l'alarme du MPU Série B	17
4.2.	Protocole ModBus – Exemple d'échange de messages	18
4.2.1.	Exemple de message de lecture ModBus.....	18
4.2.2.	Exemple de message d'écriture ModBus.....	19
5.	LOGICIEL DE COMMUNICATION WINDOWS	20
5.1.	Bibliothèque de liens dynamiques MPUCOMM	20
5.1.1.	Installation.....	20
5.1.2.	Exécution des programmes de test.....	21
5.1.3.	Interface de programmation réseau	22
5.1.4.	Codes de retour d'état	26
5.2.	Logiciel WinScreen du MPU Série B	26

FIGURES

Figure 1 - Exemple d'application du compteur électronique	16
Figure 2 - Exemple de message de lecture ModBus.....	18
Figure 3 - Exemple de message de réponse à la lecture ModBus.....	18
Figure 4 - Exemple de message d'écriture ModBus.....	19
Figure 5 - Exemple de message de réponse à l'écriture ModBus	19
Figure 6 - Interface utilisateur de DLLTester.....	21
Figure 7 - Exécution de l'application SimpleTalk.exe.....	22

1. INTRODUCTION

1.1. GENERALITÉS

Ce document décrit comment utiliser les interfaces ModBus des MPU Série B / Série Ultra. L'interface série et l'interface Ethernet sont présentées.

1.2. Références

	N° doc.	Titre
1	USM-0000020565 (MNKS001)	Manuel de l'utilisateur du MPU Série B : <i>Manuel de l'utilisateur – Configuration des ports série ModBus.</i>
2		Andy Swales, Schneider Electric: <i>Open ModBus/TCP Specification</i> , Version 1.0 29 mars 1999

1.3. Abréviations

CP	Processeur de communication
DSP	Processeur de signal numérique
VOS	Vitesse du son

1.4. Définitions

IP	Protocole Internet
MPU	Compteur ultra-sons à trajectoires multiples
TCP	Protocole de contrôle de transmissions

Cette page a été laissée vierge à dessein

2. DETAILS DU PROTOCOLE

Le protocole ModBus du MPU Série B/Série Ultra fonctionne uniquement en mode RTU.

Tous les nombres à virgule flottante sont des nombres réels codés sur 32 bits dans deux registres ModBus consécutifs codés sur 16 bits.

Tous les nombres entiers sont codés sur 32 bits dans deux registres ModBus consécutifs codés sur 16 bits.

Le MPU Série B/Série Ultra agit en tant qu'esclave ModBus.

2.1. ModBus sur port série

Le MPU Série B/Série Ultra prend en charge les connexions ModBus sur un ou deux ports série.

2.1.1. Port série 1

Les paramètres de ce port sont indiqués dans le tableau ci-dessous.

Débit en bauds	9600
Parité	Aucune
Bits d'information	8
Bits d'arrêt	1
Numéro de nœud de ModBus	1

2.1.2. Port série 2

Les paramètres suivants sont définis pour le port 2 :

Parité	Aucune
Bits d'information	8
Bits d'arrêt	1

Le reste des paramètres est configurable sur ce port.

2.1.3. Détails de l'interface matérielle

L'interface matérielle actuellement utilisée peut être l'une des interfaces suivantes :

1. RS-232 (Duplex intégral)
2. Liaison à 2 fils RS-485 (Semi-duplex)
3. Liaison à 4 fils RS-485, à savoir RS-422 (Duplex intégral)

2.1.4. Délais à respecter pour RS-485

Pour l'interface en semi-duplex RS-485 (à 2 fils), les délais suivants doivent être respectés par le logiciel sur le maître ModBus :

Un « intervalle silencieux » d'au moins 20 ms à compter de la réception d'une réponse ModBus du MPU doit exister avant que le message de requête ModBus suivant ne soit émis par le maître à l'intention du MPU. Le logiciel sur le maître ModBus doit généralement suivre ce modèle de communication :

1. Envoi d'un message de requête A au compteur
2. Lecture du message de réponse du compteur
3. **Patienter pendant au moins 20 ms**
4. Envoi du prochain message de requête B au compteur
...etc.

Cela permet de garantir que le compteur dispose de suffisamment de temps pour modifier les instructions du pilote avant que le maître envoie la requête suivante.

2.2. ModBus sur TCP/IP

Le ModBus du MPU Série B/Série Ultra au niveau de l'implémentation TCP/IP est conforme à la définition de la section 1.2 Références [2].

Les exigences suivantes s'appliquent à l'en-tête du *Protocole d'application ModBus* (MBAP) utilisé pour TCP/IP :

Champs	Longueur	Client
ID de transaction	2 octets	Doit être 0
Identifiant de protocole	2 octets	Doit être 0
Longueur	2 octets	Minimum 6
ID d'unité	1 octet	Doit être 1

3. MAPPAGE DE REGISTRES

Cette section énumère les numéros d'objet de la base de données et les adresses ModBus spécifiques au MPU Série B/Série Ultra.

Le listage suivant contient les objets les plus importants disponibles dans le protocole MODBUS. Pour une liste complète des objets, veuillez utiliser le rapport de la base de données dans le logiciel WinScreen.

Présentation générale de l'objet :

Numéro de l'objet	Adresse ModBus	Description	Format
0-199	0-398	Valeurs mesurées/calculées (lecture seule)	Float32
200-299	400-598	Alarme / État / Accumulateurs / Informations de version (lecture seule)	Int32
400-699	800-1398	Paramètres (limités). Utiliser WinScreen	Float32
700-799	1400-1598	Modes (limités). Utiliser WinScreen	Int32
1000-1100	2000-2198	Registres d'entrée externes (écrire) (P / T / Z / Densité / GC)	Float32

Dans le cas d'un compteur à 4/3/1 trajectoire(s), les numéros d'objet /adresses ModBus sont identiques, et les objets avec plus de trajectoires ne sont pas applicables.

3.1. Informations de version

Les registres suivants contiennent les informations de version du compteur. Tous les registres sont en lecture seule et des nombres entiers codés sur 32 bits.

Numéro de l'objet	Adresse ModBus	Description
250	500	Numéro de série UDSP
251	502	Numéro de série UAFE
252	504	Numéro de série UACF
264	528	Mode de protection MID (0=désactivé, 1=MID)
265	530	Verrouillage matériel (0=NA, 1=Déverrouillé, 2=Verrouillé)
266	532	Total de contrôle de la base de données de tous les paramètres liés aux poids et mesures
271	542	Total de contrôle de référence de la base de données (enregistré)
272	544	Compteur d'événements
273	546	Total de contrôle du logiciel
274	548	Version du logiciel

3.2. Registres de résultats/de sortie

Les registres suivants sont les registres de résultats du MPU Série B. Ce sont tous des nombres à virgule flottante codés sur 32 bits. Ils sont mis à jour à la fin de chaque cycle de mesure.

Numéro de l'objet	Adresse ModBus	Description	Unité métrique	Unité US
0	0	Nombre de journaux ^{*1}	-	-
1	2	État de l'alarme ^{*2}	-	-
2	4	Vitesse de débit	m/s	pied/s
3	6	Vitesse du son	m/s	pied/s
4	8	Débit volumétrique réel	m ³ /h	pied ³ /h ^{*4} baril/h ^{*5}
5	10	Volume direct accumulé	m ³	pied ³ ^{*4} baril ^{*5}
6	12	Volume inverse accumulé	m ³	pied ³ ^{*4} baril ^{*5}
7	14	Planéité du profil (tous sauf compteurs mono-trajectoire)	%	%
8	16	Symétrie du profil (tous sauf compteurs mono-trajectoire)	%	%
9	18	Flux de tourbillon (uniquement compteurs à 6 trajectoires) Volume standard accumulé, direct (MPU à 4 trajectoires et moins)	% m ² ³	% pied carré ³ ^{*4}
10	20	Flux transversal (uniquement compteurs à 6 trajectoires) Volume standard accumulé, inverse (MPU à 4 trajectoires et moins)	% m ² ³	% pied carré ³ ^{*4}
11	22	Durée d'incrémentation ^{*3}	s	s
12	24	Pression de fluide utilisée	barA	psiA
13	26	Température de ligne utilisée	°C	°F
14	28	Vitesse du débit mesurée trajectoire 1	m/s	pied/s
15	30	Vitesse du débit mesurée trajectoire 2	m/s	pied/s
16	32	Vitesse du débit mesurée trajectoire 3	m/s	pied/s
17	34	Vitesse du débit mesurée trajectoire 4	m/s	pied/s
18	36	Vitesse du débit mesurée trajectoire 5	m/s	pied/s
19	38	Vitesse du débit mesurée trajectoire 6	m/s	pied/s

MPU Série B et Série Ultra

Débitmètre ultra-sons

Numéro de l'objet	Adresse ModBus	Description	Unité métrique	Unité US
20	40	Vitesse du son mesurée trajectoire 1	m/s	pied/s
21	42	Vitesse du son mesurée trajectoire 2	m/s	pied/s
22	44	Vitesse du son mesurée trajectoire 3	m/s	pied/s
23	46	Vitesse du son mesurée trajectoire 4	m/s	pied/s
24	48	Vitesse du son mesurée trajectoire 5	m/s	pied/s
25	50	Vitesse du son mesurée trajectoire 6	m/s	pied/s
26	52	Pourcentage des signaux utilisés 1A	%	%
27	54	Pourcentage des signaux utilisés 1B	%	%
28	56	Pourcentage des signaux utilisés 2A	%	%
29	58	Pourcentage des signaux utilisés 2B	%	%
30	60	Pourcentage des signaux utilisés 3A	%	%
31	62	Pourcentage des signaux utilisés 3B	%	%
32	64	Pourcentage des signaux utilisés 4A	%	%
33	66	Pourcentage des signaux utilisés 4B	%	%
34	68	Pourcentage des signaux utilisés 5A	%	%
35	70	Pourcentage des signaux utilisés 5B	%	%
36	72	Pourcentage des signaux utilisés 6A	%	%
37	74	Pourcentage des signaux utilisés 6B	%	%
38	76	Gain 1A	-	-
39	78	Gain 1B	-	-
40	80	Gain 2A	-	-
41	82	Gain 2B	-	-
42	84	Gain 3A	-	-
43	86	Gain 3B	-	-
44	88	Gain 4A	-	-
45	90	Gain 4B	-	-

Numéro de l'objet	Adresse ModBus	Description	Unité métrique	Unité US
46	92	Gain 5A	-	-
47	94	Gain 5B	-	-
48	96	Gain 6A	-	-
49	98	Gain 6B	-	-
50	100	Rapport signal à bruit Brut 1	dB	dB
51	102	Rapport signal à bruit Brut 2	dB	dB
52	104	Rapport signal à bruit Brut 3	dB	dB
53	106	Rapport signal à bruit Brut 4	dB	dB
54	108	Rapport signal à bruit Brut 5	dB	dB
55	110	Rapport signal à bruit Brut 6	dB	dB
56	112	Rapport signal à bruit utilisé 1	dB	dB
57	114	Rapport signal à bruit utilisé 2	dB	dB
58	116	Rapport signal à bruit utilisé 3	dB	dB
59	118	Rapport signal à bruit utilisé 4	dB	dB
60	120	Rapport signal à bruit utilisé 5	dB	dB
61	122	Rapport signal à bruit utilisé 6	dB	dB
62	124	Niveau de turbulence 1	%	%
63	126	Niveau de turbulence 2	%	%
64	128	Niveau de turbulence 3	%	%
65	130	Niveau de turbulence 4	%	%
66	132	Niveau de turbulence 5	%	%
67	134	Niveau de turbulence 6	%	%

*1 Il s'agit d'un compteur incrémenté par un à la fin de chaque cycle de mesure.

*2 Il s'agit d'un message sur l'état de l'alarme. Une valeur non nulle indique ici qu'une ou plusieurs alarmes ont été déclenchées sur le compteur. Voir le chapitre 4.1 pour plus de détails.

*3 Il s'agit de la durée écoulée pendant le cycle de mesure le plus récent

*4 Compteurs de gaz

*5 Compteurs pour liquide

3.3. Accumulateurs haute résolution

Pour le transfert de données volumétriques, les accumulateurs haute résolution doivent être utilisés. Ils disposent d'une résolution très élevée et fixe, afin de ne perdre aucun volume en raison de l'arrondissement des valeurs. N'utilisez pas les objets 5 et 6 pour le transfert de volume. Ils sont indiqués uniquement à titre d'information. Les erreurs d'arrondi au niveau des nombres à virgule flottante codés sur 32 bits peuvent être importantes.

Utilisez ces accumulateurs haute résolution pour calculer les incréments de volume en lisant un certain intervalle et en soustrayant le volume :

Incrément de volume = (volume actuel - volume précédent).

Tous les objets sont des nombres entiers codés sur 32 bits et doivent être utilisés par paire.

Pour convertir le volume à l'aide d'un nombre à virgule flottante codé sur 64 bits, utilisez la formule suivante :

Volume = partie entière + partie fractionnaire * 1.0E-9

Numéro de l'objet	Adresse ModBus	Description	Unité métrique	Unité US
230	460	Volume direct accumulé Partie entière	m ³	pied ³ *1 baril*2
231	462	Volume direct accumulé Partie fractionnaire (1.0E-9)		
232	464	Volume inverse accumulé Partie entière	m ³	pied ³ *1 baril*2
233	466	Volume inverse accumulé Partie fractionnaire (1.0E-9)		
234	468	Erreur de volume direct accumulé Partie entière	m ³	pied ³ *1 baril*2
235	470	Erreur de volume direct accumulé Partie fractionnaire (1.0E-9)		
236	472	Erreur de volume inverse accumulé Partie entière	m ³	pied ³ *1 baril*2
237	474	Erreur de volume inverse accumulé Partie fractionnaire (1.0E-9)		
238	476	Volume standard accumulé direct Partie entière	m ³	pied carre ³ *1
239	478	Volume standard accumulé direct Partie fractionnaire (1.0E-9)		
240	480	Volume standard accumulé inverse Partie entière	m ³	pied carre ³ *1
241	482	Volume standard accumulé direct Partie fractionnaire (1.0E-9)		

*1 Compteurs de gaz

*2 Compteurs pour liquide

3.4. Résultats de calcul DECA/GERG (MPU)

Cette section contient les résultats des calculs DECA/GERG :

Numéro de l'objet	Adresse ModBus	Description	Unité
170	340	Vitesse du son calculée	m/s
171	342	Poids molaire moyen calculé *1	kg/mol
172	344	DZ/DP calculé *1	1/Pa
173	346	DZ/DT calculé *1	1/K
174	348	Capacité thermique isobare calculée (DECA) –OU- Pouvoir calorifique supérieur (GERG)	J/kmol (DECA) MJ/m3 (GERG)
175	350	Compressibilité du gaz calculée, ligne	-
176	352	Densité de gaz calculée, ligne	kg/m3
177	354	Compressibilité du gaz calculée, référence *1	
178	356	K, rapport de compressibilité *1	
179	358	Pouvoir calorifique inférieur	MJ/m3
180	360	Mole de méthane (C1) *1	% mol
181	362	Mole d'azote (N2) *1	% mol
182	364	Mole de dioxyde de carbone (CO2) *1	% mol
183	366	Mole d'éthane (C2) *1	% mol
184	368	Mole de propane (C3) *1	% mol
185	370	Mole d'eau (H2O) *1	% mol
186	372	Mole de sulfure d'hydrogène (H2S) *1	% mol
187	374	Mole d'hydrogène (H2) *1	% mol
188	376	Mole d'oxyde de carbone (CO) *1	% mol
189	378	Mole d'oxygène (O2) *1	% mol
190	380	Mole d'isobutane (IC4) *1	% mol
191	382	Mole de n-butane (NC4) *1	% mol
192	384	Mole d'isopentane (IC5) *1	% mol
193	386	Mole de n-pentane (NC5) *1	% mol
194	388	Mole de n-hexane (NC6) *1	% mol
195	390	Mole de n-heptane (NC7) *1	% mol
196	392	Mole de n-octane (NC8) *1	% mol

*1 DECA uniquement (pas GERG)

REMARQUE : Applicable uniquement pour les compteurs de gaz (MPU)

REMARQUE : Toutes les valeurs sont des nombres à virgule flottante codés sur 32 bits.

3.5. Registres d'entrée

Les registres suivants doivent être mis à jour en continu par l'appareil externe.

La température et la pression (objets 1000 et 1001) sont utilisées pour corriger le diamètre intérieur, les longueurs et les angles de trajectoires. Elles sont également utilisées pour sélectionner le nœud de calibrage des transducteurs.

La densité et la compressibilité (objets 1002-1005) sont utilisées pour calculer le débit massique et le débit volumétrique standard. (MPU uniquement)

L'écriture dans ces registres peut être évitée si ces fonctions ne sont pas nécessaires, ou si la température et la pression sont obtenues par interface analogique.

AVERTISSEMENT : Les objets de la série 1000 tels que ceux cités sont les seuls registres pouvant être mis à jour en continu ; la mise à jour d'autres paramètres remplira le journal des modifications et sollicitera inutilement les mémoires non volatiles.

Numéro de l'objet	Adresse ModBus	Description	Unité métrique	Unité US
1000	2000	Pression de fluide, source externe	barA	psiA
1001	2002	Température de ligne, source externe	°C	°F
1002 ^{*1}	2004	Densité du gaz dans les conditions de la ligne, source externe	Kg/m ³	livre/pied ³
1003 ^{*1}	2006	Densité du gaz dans les conditions de référence, source externe	Kg/m ² ³	livre/pied carré ³
1004 ^{*1}	2008	Compressibilité du gaz dans les conditions de la ligne, source externe	-	
1005 ^{*1}	2010	Compressibilité du gaz dans les conditions de référence, source externe	-	
1006 ^{*1}	2012	Mole de méthane (C1) ^{*2}	% mol	% mol
1007 ^{*1}	2014	Mole d'azote (N2)	% mol	% mol
1008 ^{*1}	2016	Mole de dioxyde de carbone (CO2)	% mol	% mol
1009 ^{*1}	2018	Mole d'éthane (C2) ^{*2}	% mol	% mol
1010 ^{*1}	2020	Mole de propane (C3) ^{*2}	% mol	% mol
1011 ^{*1}	2022	Mole d'eau (H2O) ^{*2}	% mol	% mol
1012 ^{*1}	2024	Mole de sulfure d'hydrogène (H2S) ^{*2}	% mol	% mol
1013 ^{*1}	2026	Mole d'hydrogène (H2) ^{*2}	% mol	% mol
1014 ^{*1}	2028	Mole d'oxyde de carbone (CO) ^{*2}	% mol	% mol

1015 ^{*1}	2030	Mole d'oxygène (O2) ^{*2}	% mol	% mol
1016 ^{*1}	2032	Mole d'isobutane (IC4) ^{*2}	% mol	% mol
1017 ^{*1}	2034	Mole de n-butane (NC4) ^{*2}	% mol	% mol
1018 ^{*1}	2036	Mole d'isopentane (IC5) ^{*2}	% mol	% mol
1019 ^{*1}	2038	Mole de n-pentane (NC5) ^{*2}	% mol	% mol
1020 ^{*1}	2040	Mole de n-hexane (NC6) ^{*2}	% mol	% mol
1021 ^{*1}	2042	Mole de n-heptane (NC7) ^{*2}	% mol	% mol
1022 ^{*1}	2044	Mole de c-octane (NC8) ^{*2}	% mol	% mol
1023 ^{*1}	2046	Mole de n-nonane (NC9) ^{*2}	% mol	% mol
1024 ^{*1}	2048	Mole de n-décane (NC10) ^{*2}	% mol	% mol
1025 ^{*1}	2050	Mole d'hélium (HE) ^{*2}	% mol	% mol
1026 ^{*1}	2052	Mole d'argon (AR) ^{*2}	% mol	% mol
1027 ^{*1}	2054	Mole d'hexane+ (C6+) ^{*2}	% mol	% mol
1028 ^{*1}	2056	Mole rest. ^{*2}	% mol	% mol

^{*1} Compteurs de gaz uniquement (MPU)

^{*2} DECA uniquement (pas GERG)

REMARQUE : Toutes les valeurs sont des nombres à virgule flottante codés sur 32 bits.

4. UTILISATION DU REGISTRE

La Figure 1 ci-dessous décrit la manière la plus fréquente d'implémenter une application maître synchronisée avec le compteur.

Dans cet exemple, l'application maître exécutée sur un compteur électronique externe est responsable de l'écriture de la pression et de la température correctes vers le compteur.

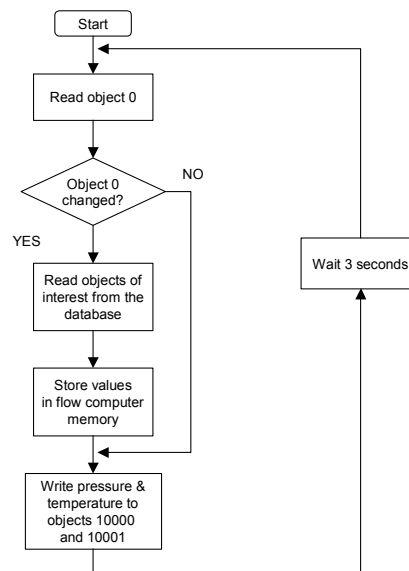


Figure 1 - Exemple d'application du compteur électronique

MPU Série B et Série Ultra

Débitmètre ultra-sons

Le maître peut également lire un bloc complet de données à partir de l'objet 0 pour simplifier les opérations ModBus et réduire leur nombre. Chaque fois que l'objet 0 modifie des valeurs, des valeurs sont mises à jour dans le bloc.

4.1. État de l'alarme du MPU Série B

L'état de l'alarme du MPU Série B est un nombre codé sur des bits indiquant l'état des alarmes du MPU Série B. Pour interpréter correctement les bits, le nombre à virgule flottante codé sur 32 bits doit être converti en un nombre entier codé sur 32 bits. Le tableau ci-dessous indique les bits d'alarme utilisés et leur interprétation.

Bit	Valeur	Interprétation de l'alarme
0	1	Alarme de débit élevé
1	2	Défaillance du matériel
2	4	Défaillance du transducteur
3	8	Échec de calcul
4	16	Faible pourcentage de rafales
5	32	Erreur de gain
6	64	Écart trop important de la vitesse du son
7	128	Corrections du débit actives
8	256	Erreur de paramètre
9	512	Faible rapport signal à bruit
10	1024	Niveau de turbulence élevé
11	2048	Écart important du profil

4.2. Protocole ModBus – Exemple d'échange de messages

Ce chapitre décrit l'échange de messages se produisant dans une communication compteur électronique - MPU typique.

4.2.1. Exemple de message de lecture ModBus

Dans l'exemple suivant, le compteur électronique effectue la tâche suivante :

- Lecture de la vitesse du son et du débit depuis le MPU Série B

Les objets de la base de données utilisés pour la vitesse du son et le débit ont les numéros d'objet 3 et 4. Ceux-ci ont les adresses ModBus 6 (07D0 Hex) et 8. Ces registres sont lus avec UN message dont le code fonctionnel est 3. La figure 2 ci-dessous indique le contenu de ce message.



Figure 2 - Exemple de message de lecture ModBus

Remarque : Tous les octets sont indiqués au format hexadécimal.

La réponse du MPU Série B sera au format décrit dans la Figure 3 ci-dessous.

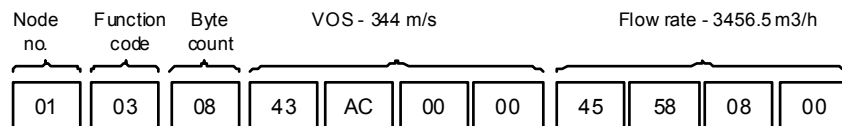


Figure 3 - Exemple de message de réponse à la lecture ModBus

Le message comprend le contenu des objets 3 (vitesse du son) et 4 (débit) de la base de données, représentés sous forme de deux nombres réels codés sur 32 bits (l'octet le moins important en premier).

4.2.2. Exemple de message d'écriture ModBus

Dans cet exemple, l'ordinateur électronique effectue la tâche suivante :

- Écriture de la température et de la pression depuis le MPU Série B

Les objets de la base de données utilisés pour la pression et la température à partir d'une source externe sont les numéros d'objet 1000 et 1001. Ceux-ci ont les adresses ModBus 2000 (07D0 Hex) et 2002. Ces registres sont écrits avec UN message dont le code fonctionnel est 16. La Figure 4 ci-dessous indique le contenu de ce message.

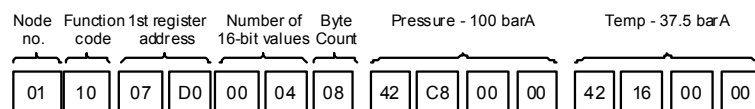


Figure 4 - Exemple de message d'écriture ModBus

La réponse du MPU Série B sera au format décrit dans la figure ci-dessous.



Figure 5 - Exemple de message de réponse à l'écriture ModBus

Ce message contient une copie des 6 premiers octets du message de requête.

5. LOGICIEL DE COMMUNICATION WINDOWS

5.1. Bibliothèque de liens dynamiques MPUCOMM

Cette ligne de communication fournit une méthode simple et rapide de communication avec le MPU Série B à l'aide de l'interface réseau. L'interface est basée sur une bibliothèque de liens dynamiques Microsoft Windows. Cette bibliothèque de liens dynamiques encapsule l'ensemble des communications avec le MPU Série B et offre un jeu de fonctions de bibliothèque pouvant être appelées depuis votre logiciel. (Les bibliothèques de liens dynamiques peuvent être appelées dans la plupart des langages de programmation, y compris Microsoft Visual C++, Microsoft Visual Basic, National Instrument LabView/LabWindows, applications Office pouvant être programmées avec des macros Visual Basic, à savoir Excel).

L'OP pour utiliser cette bibliothèque de liens dynamiques doit contenir les « éléments » suivants :

1. Exécution sur Microsoft Windows 98/NT//2000 ou versions ultérieures.
2. Réseau Ethernet (ou une carte Ethernet dans votre ordinateur)
3. Protocole TCP/IP installé sous Windows.
4. L'application peut ensuite appeler une DLL (Bibliothèque de liens dynamiques) Windows. La DLL contient des fonctions simples pour lire ou écrire dans la base de données du MPU Série B via le réseau.

5.1.1. Installation

L'adresse IP du MPU Série B doit être définie dans le fichier hôte. Par exemple :

128.1.221.121 MPUsn11

où « 128.1.221.121 » est l'adresse IP, et « MPUsn11 » est le nom sous lequel la machine peut être référencée.

Ce fichier se trouve généralement dans le répertoire C:\WINNT\system32\drivers\etc\ sur les OP fonctionnant sous Windows NT, et dans le répertoire C:\WINDOWS\... sur les OP fonctionnant sous Windows 98.

Les fichiers suivants sont inclus dans la boîte à outils de communication réseau du MPU Série B :

1. *MPUBComm.dll* - Bibliothèque callable contenant l'implémentation du système de communication.
2. *MPUBComm.lib* - Bibliothèque de liens pour Microsoft Visual C++.
3. *DLLTester.exe* – Un programme de test exécutable comportant une interface utilisateur simple
4. *SimpleTalk.cpp* – Un exemple de programme de travail très simple en C indiquant l'utilisation de la DLL.
5. *MPUBCommExample.exe* - Fichier exécutable des fonctions ci-dessus.
6. *W95ws2setup.exe* – Installation de la bibliothèque WinSock version 2 pour Windows 95.

Remarque : La bibliothèque utilise WinSock 2. Ce composant est standard sous Windows NT 4.0 et Windows 98. Il n'est pas installé sur certaines versions de Windows 95. Si vous rencontrez des problèmes sous Windows 95, exécutez le programme « W95ws2setup.exe ». Il installera WinSock 2.



Avertissement : N'exécutez pas « W95ws2setup.exe » si vous possédez Windows NT 4.0, Windows 98 ou une version ultérieure.

5.1.2. Exécution des programmes de test

Les deux programmes de test doivent être exécutés à partir du même répertoire dans lequel le fichier *MPUBComm.dll* se trouve.

DLLTester.exe

Ce programme de test affiche une boîte de dialogue simple dans laquelle l'utilisateur peut lire et écrire des valeurs dans la base de données du MPU Série B. Voir la Figure 6 ci-dessous.

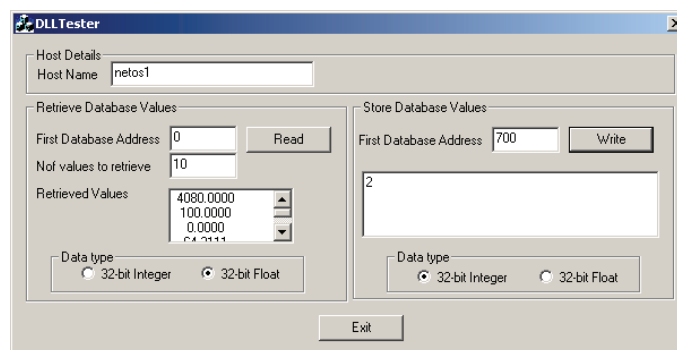


Figure 6 - Interface utilisateur de DLLTester

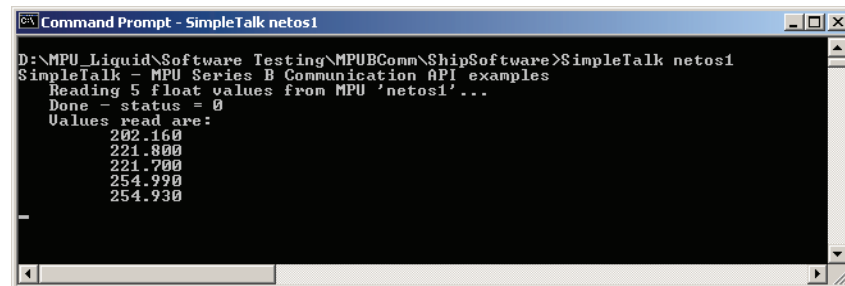
SimpleTalk.exe

Il s'agit d'une application de console simple qui lit 5 valeurs dans la base de données du MPU Série B. Elle lit toujours les 5 mêmes objets, en commençant par le numéro d'objet 0 de la base de données.

Ouvrez une fenêtre DOS et saisissez la commande suivante :

```
SimpleTalk <machine name>
```

La Figure 7 illustre un modèle d'exécution de cette application.



```
Command Prompt - SimpleTalk netos1
D:\MPU_Liquid\Software_Testing\MPUBComm\ShipSoftware>SimpleTalk netos1
SimpleTalk - MPU Series B Communication API examples
Reading 5 float values from MPU 'netos1' ...
Done - status = 0
Values read are:
    202.160
    221.800
    221.700
    254.990
    254.930
```

Figure 7 - Exécution de l'application SimpleTalk.exe

5.1.3. Interface de programmation réseau

Les fonctions suivantes sont implémentées dans la DLL :

GetFloatValues

Cette fonction lit l'objet sous forme de nombres à virgule flottante dans la base de données du MPU Série B.

```
int DLLAPI getFloatValues(
    // Input
    const char *lpszHost, //
    Hostname to connect to (MPUsn11)
    unsigned short nFirstObjectNumber, //
    First database object to read
    unsigned short nNOofObject, // The
    number of objects to read
    // Output
    float *pfValueBuffer //
    Buffer to store the values
);
```

Paramètres :

- **LpszHost**
Spécifie l'hôte à connecter. L'hôte doit être défini dans le fichier hôte sur votre OP. La chaîne doit se terminer par un Null.
- **NFirstObjectNumber**
Spécifie le numéro d'objet de la première valeur de la base de données à lire. Ces numéros d'objet sont définis dans la section 3.
- **NNOfObject**
Spécifie le nombre d'objets à lire. Il doit s'agir d'un nombre compris entre 1 et 64. Le paramètre doit être un nombre entier codé sur 16 bits.
- **PfValueBuffer**
Spécifie l'adresse mémoire à laquelle la fonction placera les valeurs lues. Le paramètre doit être un pointeur vers une matrice de nombres à virgule flottante en simple précision, codés sur 32 bits. La taille de la matrice doit être au moins égale à `nFirstObjectNumber * 4` octets.

GetIntValues

Cette fonction lit l'objet sous forme de nombres à virgule flottante dans la base de données du MPU Série B.

```
int DLLAPI getFloatValues(  
    // Input  
    const char    *lpszHost,           //  
    Hostname to connect to (MPUsn11)  
    unsigned short nFirstObjectNumber, //  
    First database object to read  
    unsigned short nNOfObject,        //  
    number of objects to read  
    // Output  
    int           *pnValueBuffer      //  
    Buffer to store the values  
);
```

Paramètres :

- **LpszHost**
Spécifie l'hôte à connecter. L'hôte doit être défini dans le fichier hôte sur votre OP. La chaîne doit se terminer par un Null.
- **NFirstObjectNumber**
Spécifie le numéro d'objet de la première valeur de la base de données à lire. Ces numéros d'objet sont définis dans la section 3.

- **NNOfObject**
Spécifie le nombre d'objets à lire. Il doit s'agir d'un nombre compris entre 1 et 64. Le paramètre doit être un nombre entier codé sur 16 bits.
- **PnValueBuffer**
Spécifie l'adresse mémoire à laquelle la fonction placera les valeurs lues. Le paramètre doit être un pointeur vers une matrice de nombres entiers codés sur 64 bits. La taille de la matrice doit être au moins égale à `nFirstObjectNumber * 8` octets.

SetFloatValues :

Cette fonction écrit les nombres à virgule flottante dans la base de données du MPU Série B.

```
int DLLAPI setFloatValues(  
    // Input  
    const char      *lpszHost,           //  
    Hostname to connect to (MPUsn11)  
    unsigned short nFirstObjectNumber, //  
    First database object to read  
    unsigned short nNOfObject,         // The  
    number of objects to read  
    float          *pfValueBuffer      //  
    Buffer with the values  
);
```

Paramètres :

- **LpszHost**
Spécifie l'hôte à connecter. L'hôte doit être défini dans le fichier hôte sur votre OP. La chaîne doit se terminer par un Null.
- **NFirstObjectNumber**
Spécifie le numéro d'objet de la première valeur de la base de données à écrire. Ces numéros d'objet sont définis dans la section 3.
- **NNOfObject**
Spécifie le nombre d'objets à écrire. Il doit s'agir d'un nombre compris entre 1 et 64. Le paramètre doit être un nombre entier codé sur 16 bits.
- **PfValueBuffer**
Spécifie l'adresse mémoire d'où sont extraites les valeurs à écrire. Le paramètre doit être un pointeur vers une matrice de nombres en simple précision codés sur 32 bits. La taille de la matrice doit être au moins égale à `nFirstObjectNumber * 4` octets.

setIntValues

Cette fonction écrit les nombres entiers dans la base de données du MPU Série B.

```
int DLLAPI setIntValues(  
    // Input  
    const char      *lpszHost,           //  
    Hostname to connect to (MPUsn11)  
    unsigned short  nFirstObjectNumber, //  
    First database object to read  
    unsigned short  nNOofObject,       // The  
    number of objects to read  
    int             *pnValueBuffer     //  
    Buffer with the values  
);
```

Paramètres :

- **LpszHost**
Spécifie l'hôte à connecter. L'hôte doit être défini dans le fichier hôte sur votre OP. La chaîne doit se terminer par un Null.
- **nFirstObjectNumber**
Spécifie le numéro d'objet de la première valeur de la base de données à écrire. Ces numéros d'objet sont définis dans la section 3.
- **nNOofObject**
Spécifie le nombre d'objets à écrire. Il doit s'agir d'un nombre compris entre 1 et 64. Le paramètre doit être un nombre entier codé sur 16 bits.
- **pnValueBuffer**
Spécifie l'adresse mémoire d'où sont extraites les valeurs à écrire. Le paramètre doit être un pointeur vers une matrice de nombres entiers codés sur 64 bits. La taille de la matrice doit être au moins égale à `nFirstObjectNumber * 8` octets.

5.1.4. Codes de retour d'état

Cette fonction implémentée dans le DLL retourne un code de retour d'état unique. Ce code est retourné sous forme d'un nombre entier codé sur 64 bits.

Le tableau ci-dessous indique les valeurs retournées valides.

Code de retour d'état	Description
0	L'appel de la fonction a réussi
1	L'appel de la fonction a échoué
2	L'essai de connexion au MPU Série B a échoué
3	L'adresse de la base de données spécifiée est invalide

5.2. Logiciel WinScreen du MPU Série B

Le logiciel WinScreen du MPU Série B est une application autonome permettant de communiquer avec le MPU Série B. Il peut être utilisé aussi bien avec l'interface TCP/IP qu'avec l'interface série ModBus. Reportez-vous au « Manuel de l'utilisateur, MPU Série B » (MNKS001) PRD-0000020565 pour plus de détails.

Cette page a été laissée vierge à dessein.

Les spécifications figurant aux présentes sont susceptibles d'être modifiées sans préavis et tout utilisateur de ces spécifications doit vérifier auprès du fabricant que les spécifications sont actuellement en vigueur. Sinon le fabricant décline toute responsabilité pour l'utilisation de spécifications qui ont été modifiées et ne sont plus en vigueur.

Les coordonnées sont susceptibles d'être modifiées. Pour obtenir nos coordonnées les plus récentes, consultez notre site Web sur www.fmctechnologies.com/measurementssolutions et cliquez sur le lien « Contact Us » dans la colonne de gauche.

Siège social :

500 North Sam Houston Parkway West, Suite 100, Houston, TX 77067 USA, Téléphone : +1 (281) 260 2190, Fax : +1 (281) 260 2191

Produits et équipements de mesure :

Erie, PA États-Unis d'Amérique +1 (814) 898 5000

Eilerbek, Allemagne +49 (4101) 3040

Barcelone, Espagne +34 (93) 201 0989

Pékin, Chine +86 (10) 6500 2251

Buenos Aires, Argentine +54 (11) 4312 4736

Burnham, Angleterre +44 (1628) 603205

Dubaï, Émirats Arabes Unis +971 (4) 883 0303

Los Angeles, CA États-Unis d'Amérique +1 (310) 328 1236

Melbourne, Australie +61 (3) 9807 2818

Moscou, Russie +7 (495) 5648705

Singapour, +65 6861 3011

Thetford, Angleterre +44 (1842) 822900

Systèmes de mesure intégrés :

Corpus Christi, TX États-Unis d'Amérique +1 (361) 289 3400

Kongsberg, Norvège +47 (32) 286700

Dubaï, Émirats Arabes Unis +971 (4) 883 0303

Visitez notre site Web sur www.fmctechnologies.com/measurementssolutions