



LSI LASTEM S.r.l.

Via Ex S.P. 161 Dosso, n.9 - 20090 Settala Premenugo (MI) - Italia

Tel.: (+39) 02 95 41 41

Fax: (+39) 02 95 77 05 94

e-mail: info@lsi-lastem.it

WEB: <http://www.lsi-lastem.it>

CF./P. Iva: (VAT) IT-04407090150

REA:1009921 **Reg.Imprese:** 04407090150



Datalogger



Protocoles de communication

Mise à jour 20/10/2011

Copyright 2010-2011 LSI LASTEM. Tous les droits sont réservés.

Les informations contenues dans ce manuel sont susceptibles d'être modifiées sans préavis.
Aucune partie de ce manuel ne peut être reproduite sous n'importe quelle forme ou par n'importe quels moyens électroniques ou mécaniques, et dans quelque but que ce soit, sans le consentement écrit de LSI LASTEM.

LSI LASTEM se réserve le droit de modifier le produit, sans être obligé de mettre à jour rapidement ce document.

Sommaire

1	Introduction.....	4
2	Protocole TTY	4
2.1	Description des messages.....	4
2.2	Format des messages d’interrogation.....	4
2.2.1	Mise à jour de l’heure de l’instrument	5
2.3	Format des messages de réponse	6
2.3.1	Calcul du champ <i>checksum</i>	6
2.3.2	Réponse de confirmation simple.....	6
2.3.3	Valeurs instantanées.....	6
2.3.4	Données d’identification	7
2.3.5	Données diagnostiques et statistiques	7
3	Protocole Modbus	8
3.1	Format des messages.....	8
3.1.1	Adresse.....	8
3.1.2	Code fonction.....	9
3.1.3	CRC16.....	9
3.2	Fonctions supportées.....	10
3.2.1	Read Coils	10
3.2.2	Read Holding Registers	12
3.2.3	Read Input Register.....	12
3.2.4	Write Single Coil	17
3.2.5	Write Multiple Coils	19
3.2.6	Write Multiple Registers.....	20
3.2.7	Read Device Identification.....	22
3.3	Codes d’exception.....	23

1 Introduction

Les instruments E-Log et M-Log supportent plusieurs protocoles de communication utilisables par les deux ports de communication série. Ce manuel décrit les protocoles *TTY* et *Modbus* disponibles sur le port de communication n. 2.

2 Protocole TTY

À travers le protocole TTY on peut interroger l'instrument afin d'obtenir de manière simple les données échantillonnées et calculées par les mesures (valeurs instantanées) et d'autres informations utiles. La demande des valeurs mesurées est faite en mode *polling* (sur commande de l'host), ou en mode spontané avec taux de transmission réglable.

Les données sont transmises dans un format de simple interprétation, soit directement par l'opérateur à travers un programme de terminal, soit de la part de programmes de décodification ou d'importation automatique.

L'interrogation de l'instrument se fait en spécifiant dans la commande d'interrogation l'adresse de réseau; cela permet d'effectuer l'interrogation en séquence de plusieurs instruments branchés à travers une ligne de communication unique.

2.1 Description des messages

Dans les descriptions successives on utilisera les caractères < e > pour délimiter les différents champs qui constituent les messages de question et de réponse. Il faudra garder à l'esprit que ces caractères sont utilisés ici à des fins descriptifs et par conséquent ils n'apparaissent pas dans les vrais messages échangés pendant la communication.

2.2 Format des messages d'interrogation

L'envoi de la commande d'interrogation à l'instrument a lieu à travers la séquence de caractères suivante:

<A><addr><cmd><prm><CR>

où:

<A>: caractère ASCII 'A' (hexadécimal 41);

<addr>: adresse de l'instrument, exprimé par le caractère ASCII de '1' (hexadécimal 31) à '9' (hexadécimal 39); l'instrument répond uniquement s'il existe une correspondance entre l'adresse réglée à travers *3DOM* (paramètre *Indirizzo di rete/Adresse de réseau*) et cette valeur;

<cmd>: code de la commande; un caractère ASCII au choix entre:

- <I>: demande des valeurs instantanées des mesures;
- <S>: demande des données de l'instrument;

- <T>: mise à jour de l'heure (E-Log supporte cette fonction dès la version 2.01.04);
- <A>: démarrage du mode d'envoi spontané des données instantanées; l'instrument commence à transmettre les données instantanées des mesures, avec un taux temporel configurable grâce au programme *3DOM*; si ce paramètre est réglé sur zéro, l'envoi de la commande détermine la transmission automatique avec un taux fixe de 10 secondes; en cas de redémarrage de l'instrument, le mode automatique n'est pas réactivé; pour obtenir également la transmission automatique après le redémarrage de l'instrument, régler avec *3DOM* le temps de transmission automatique avec une valeur différente de zéro;
- <M>: fin du mode de transmission spontanée des données instantanées; le redémarrage de l'instrument détermine une nouvelle transmission automatique des données uniquement si un temps de transmission différent de zéro a été configuré;
- <D>: demande des données diagnostiques et statistiques de fonctionnement de l'instrument;
- <R>: mise à zéro des données diagnostiques et statistiques de fonctionnement de l'instrument; il faut remarquer que la mise à zéro comprend, outre aux données envoyées comme réponse à la commande <D>, la mise à zéro de toutes les autres informations statistiques (affichées dans les fenêtres de diagnostic; pour plus d'informations faire référence au manuel de l'instrument utilisé).

<prm>: paramètres optionnels à la demande.

<CR>: caractère ASCII *carriage return*.

Veillez noter que le redémarrage de l'instrument détermine automatiquement le démarrage de la transmission spontanée des données instantanées uniquement si le taux de transmission est réglé avec une valeur différente de zéro même sans la transmission de la commande <A>.

Toutes les commandes énumérées ci-dessus doivent être envoyées de façon complète, dans un délai pendant lequel l'instrument n'effectue pas la transmission automatique des données; en cas contraire l'instrument n'est pas capable d'interpréter la commande envoyée.

2.2.1 Mise à jour de l'heure de l'instrument

La demande de mise à jour de l'heure de l'instrument est faite à travers l'interrogation suivante:

<addr><T><P|I><yy/mm/dd hh:mm:ss>

où:

<P|I>: à utiliser en alternative; "P" indique que le réglage de l'heure par l'instrument est fait de façon progressive; "I" produit au contraire la correction de façon immédiate;

<yy/mm/dd hh:mm:ss> sont la date et l'heure qu'on doit régler; la séquence numérique doit être respectée, tandis que les séparateurs peuvent être également remplacés par des autres caractères non numériques.

2.3 Format des messages de réponse

2.3.1 Calcul du champ *checksum*

Chaque message de réponse contenant des données préparées par l'instrument comprend, en file d'attente, un champ de contrôle de la transmission (*checksum*), utile pour valider les données reçues par rapport aux données effectivement transmises.

Le champ *checksum* est contenu entre deux parenthèses carrée et est exprimé en format hexadécimal ASCII, correspondant à la somme à 8 octets en OR exclusif de tous les caractères compris entre le premier caractère et le caractère immédiatement précédant la parenthèse carrée ouverte.

2.3.2 Réponse de confirmation simple

En cas d'acceptation positive de la commande qui ne doit pas être suivie par des données (comme il arrive pour les réponses aux commandes <M> et <R>), l'instrument transmet un message de réponse contenant le texte <Ok>; en cas de rejet ou non reconnaissance de la commande reçue, ou pour toute autre erreur, le message de réponse contient le texte <Err>.

2.3.3 Valeurs instantanées

La demande des valeurs instantanées des mesures produit la transmission de la réponse suivante:

```
<addr><I><nnn.nn>;<nnn.nn>;...<[CS]><CR><LF>
```

où:

<addr>: adresse de l'instrument;

<I>: code de la commande;

<nnn.nn>: valeur instantanée (la dernière échantillonnée ou calculée) de la *n*-ième mesure, où *n* est la position de la valeur par rapport aux autres contenues dans le même message; le nombre de caractères de la partie entière est variable sur la base de la valeur prise par chaque mesure; le nombre de caractères décimaux dépend de la précision réglée pour la mesure; le nombre maximum de mesures contenues dans un unique message est 25; dans le cas où l'instrument soit programmé avec plus de 25 mesures, le premier message est suivi par d'autres messages précédés du suffixe <Mxx>; où *xx* correspond à l'indice (base 0) de la première mesure appartenant au group de mesures dans le message;

<[CS]>: champ *checksum*;

<CR>: caractère ASCII *carriage return*;

<LF>: caractère ASCII *line feed*.

2.3.4 Données d'identification

La demande des données d'identification de l'instrument produit la transmission de la réponse suivante:

```
<addr><S><M:ccc/sss>;<V:MM.mm.bb>;<S:fff/uuu><[CS]><CR><LF>
```

où:

<addr>: adresse de l'instrument;

<S>: code de la commande;

<M:ccc/sss>: modèle de l'instrument (ccc=code, sss=sous-code);

<V:MM.mm.bb>: version du firmware (MM=version supérieure, mm=version inférieure, bb=version interne);

<S:fff/uuu>: numéro de série (fff=d'usine, uuu=valeur définie par l'utilisateur).

<[CS]>: champ *checksum*;

<CR>: caractère ASCII *carriage return*;

<LF>: caractère ASCII *line feed*.

2.3.5 Données diagnostiques et statistiques

La demande des données diagnostiques et statistiques produit la transmission de la réponse suivante:

```
<addr><D><L:l1l1>;<S:sss>;<C:ccc>;<E:eee><[CS]><CR><LF>
```

où:

<addr>: adresse de l'instrument;

<D>: code de la commande;

<L:l1l1>: temps de fonctionnement ininterrompu (de vie) de l'instrument, exprimé par le format *ddd hh:mm:ss*;

<S:sss>: date/heure de la dernière mise à zéro des données statistiques, exprimé par le format *yy/mm/dd hh:mm:ss*;

<C:ccc>: date/heure du système (courant) de l'instrument, exprimé par le format *yy/mm/dd hh:mm:ss*;

<E:eee>: code hexadécimal des erreurs de fonctionnement relevées par l'instrument (pour plus d'informations faire référence au manuel de l'instrument utilisé);

<[CS]>: champ *checksum*;

<CR>: caractère ASCII *carriage return*;

<LF>: caractère ASCII *line feed*.

3 Protocole Modbus

Modbus est un protocole de communication série très utilisé dans le domaine industriel pour permettre la communication entre un *master* (généralement un PC ou un système SCADA) et un ou plus *slaves/serveurs* (équipements de mesure, de contrôle ou PLC), branchés au même réseau. Modbus définit comment le *master* et les *slaves* instaurent et interrompent la communication, comment les messages sont échangés et comment les erreurs sont détectées. C'est uniquement le *master* qui peut commencer la communication.

Une adresse univoque est assignée à chaque dispositif du réseau. Une commande Modbus contient l'adresse du dispositif destinataire vers lequel le message est adressé. Exclusivement le dispositif adressé répondra à la commande, même si les autres instruments également reçoivent le même message. Toutes les commandes Modbus contiennent des informations de contrôle, qui garantissent que la commande arrivée est correcte.

Du point de vue de Modbus, l'enregistreur de données apparaît dans le système comme un appareil *slave* et implémente une partie du protocole en version RTU.

3.1 Format des messages

Le format des messages utilisé par les dispositifs master/slave est le suivant:

<i>Nom du champ</i>	<i>Dimension</i>	<i>Description</i>
Adresse	1 byte	L'adresse du slave avec lequel le master doit communiquer (adresse de réseau de l'instrument); le message <i>broadcast</i> (ID = 0) n'est pas supporté
Code fonction	1 byte	Commande à envoyer (ou envoyée)
Données	n byte	Données à transmettre
CRC16	2 byte	Contrôle de l'erreur selon l'algorithme CRC16

Si un dispositif slave détecte une erreur dans le message reçu (erreur de format ou dans le CRC16) le message est considéré comme non-valide et rejeté, par conséquent l'action ne sera pas effectuée et aucune réponse ne sera donnée à la commande, ainsi comme dans le cas où l'adresse ne correspond pas au dispositif interrogé.

3.1.1 Adresse

L'adresse est utilisée pour identifier le destinataire du message: elle comprend l'adresse numérique du serveur sélectionné. Elle peut assumer des valeurs de 1 à 200 si utilisée dans un réseau RS-232. L'instrument compare la valeur adresse reçue avec l'adresse de réseau programmée et répond uniquement si elles sont correspondantes. Utiliser le programme *3DOM* c'est modifier le paramètre *Indirizzo di rete/Adresse de réseau* selon la nécessité.

Les messages de type *broadcast* (adresse égale à 0) ne sont pas supportés.

3.1.2 Code fonction

Le code fonction identifie la commande qui doit être envoyée ou qui vient d'être gérée par le serveur.

Les codes fonction supportés par l'instrument sont les suivants:

<i>Code fonction</i>	<i>Nom Fonction</i>	<i>Description</i>
01	Read Coils	Lit l'état des actionneurs et l'état de fonctionnement de l'instrument
03	Read Holding Registers	Comme <i>Read Input Registers</i>
04	Read Input Registers	Li les dernières valeurs acquises par l'instrument (soit mesures analogiques que numériques) et la date/heure du système
05	Write Single Coil	Règle l'état d'un actionneur de l'instrument
15	Write Multiple Coils	Mise à zéro des erreurs de fonctionnement de l'instrument
10	Write Multiple Registers	Règle la date/heure du système et quelques paramètres de configuration des mesures
43/14	Read Device Identification	Lit les informations de l'instrument

N'importe quel autre code fonction non inséré dans le tableau est ignoré par l'instrument, qui ne génère aucune réponse d'erreur par rapport à la réception d'une commande non supportée.

3.1.3 CRC16

Le CRC16 comprend le code de redondance cyclique (Cyclic Redundancy Check) calculé en utilisant l'algorithme CRC16 (polynôme $X^{16} + X^{15} + X^2 + 1$). Pour le calcul de ces deux caractères le message (adresse, code fonction et données sans les bits de démarrage, d'arrêt et la parité éventuelle) est considéré comme un numéro binaire unique continu dont le bit le plus significatif (MSB) est transmis en précedence.

3.2 Fonctions supportées

3.2.1 Read Coils

Utiliser la fonction *Read Coils* pour lire l'état des actionneurs de l'instrument et les erreurs de fonctionnement éventuelles.

La signification des valeurs assumées par les actionneurs dépend de la logique de fonctionnement des actionneurs réglée sur l'instrument (pour plus d'informations faire référence au manuel de l'instrument utilisé). Le tableau suivant en résume la signification.

Type de logique de fonctionnement	Valeur	État	Signification
À faible consommation	0	OFF	Sortie activation éteinte
	1	ON	Sortie activation allumée
En sécurité	0	ON	Sortie activation allumée
	1	OFF	Sortie activation éteinte

Pour ce qui concerne les erreurs de fonctionnement, au contraire, la valeur 0 indique *non en erreur* tandis que la valeur 1 indique *erreur*.

Adresse (Hex)	Coil	Signification
État Actionneurs		
0x00	01	État actionneur nr. 1
	02	État actionneur nr. 2
	03	État actionneur nr. 3
	04	État actionneur nr. 4
	05	État actionneur nr. 5
	06	État actionneur nr. 6
	07	État actionneur nr. 7
	08	<i>Non utilisé</i>
État fonctionnement (erreurs)		
0x01	09	Erreur Read After Write
	10	Configuration mémorisée non valide
	11	Erreur de recherche de la page de mémoire
	12	Numéro maximum de loop dépassé pendant la recherche des données en mémoire
	13	
	14	Overrun mise en attente pour l'acceptation des demandes d'acquisition
	15	Overrun mise en attente pour la mémorisation des résultats d'acquisition
	16	<i>Non utilisé</i>
0x02	17	Écriture de la page en mémoire échouée
	18	Une ou plusieurs pages de données d'élaboration perdues
	19	Effacement secteur de mémoire échoué
	20	Timeout sur l'attente de la fin de l'opération en mémoire
	21	Dispositif de mémoire non supporté
	22	Message reçu erroné (code de contrôle non valide)
	23	Message répété jusqu'à un maximum de 3 fois
	24	À la suite de la répétition maximale du message celui a été perdu
	Erreur relevée pendant l'écriture /lecture de la EEPROM à l'intérieur du serveur	

0x03	25	Le serveur n'est pas capable de opérer car sa config. n'a pas été programmée
	26	Un overflow/débordement a eu lieu sur la réception d'un message individuel
	27	Mise en attente des messages pleine dans le serveur/slave
	28	Regroupe toutes les erreurs CISS relatives à la syntaxe
	29	Erreur CISS non spécifiée (Unspecified)
	30	Erreur CISS (BadCommandCode)
	31	Erreur CISS (BadParameter)
	32	Erreur CISS (ParameterOutOfRange)
0x04	33	Erreur CISS (UnrecognizedCDV)
	34	Erreur CISS (BeyondMaxClassLevel)
	35	Erreur CISS (ParameterIndexOutOfRange)
	36	Erreur CISS (ClassIndexOutOfRange)
	37	Erreur CISS (RequestNotPermitted)
	38	<i>Non utilisé</i>
	39	<i>Non utilisé</i>
	40	<i>Non utilisé</i>

Demande

Code fonction	1 byte	0x01
Adresse de départ	2 byte	de 0x00 à 0x27
Numéro de coils	2 byte	de 1 à 40

Réponse

Code fonction	1 byte	0x01
Numéro de octets	1 byte	1
État actionneur/erreur	1 byte	1=On, 0=Off

Erreur

Codice funzione	1 byte	0x81
Code exception	1 byte	01 o 02 ou 03 ou 04

Pour plus d'informations sur *Codice eccezione/ Code exception* faire référence au chap. 3.3.

Exemple: demande de lecture de l'état des actionneurs de l'instrument avec ID égal à 01:

Demande	
Nome du champ	(Hex)
Adresse dispositif	01
Code fonction	01
Adresse de départ (Hi)	00
Adresse de départ (Lo)	00
Numéro d'actionneurs (Hi)	00
Numéro d'actionneurs (Lo)	08
CRC16 (Hi)	3D
CRC16 (Lo)	CC

Réponse à la demande avec les valeurs suivantes: 0, 0, 1, 0, 0, 0, 0, 0:

Réponse	
Nom du champ	(Hex)
Adresse dispositif	01
Code fonction	01
Numéro de octets	01
Valeur	04
CRC16 (Hi)	50
CRC16 (Lo)	4B

3.2.2 Read Holding Registers

Read Holding Register et *Read Input Register*, du point de vue de l'implémentation Modbus dans l'enregistreur de données, sont équivalents et sont pour cela traités de la même manière. Dans la demande, indiquer le code fonction 0x03(*Read Holding Register*) ou 0x04 (*Read Input Register*).

Pour plus d'informations sur ce sujet, faire référence à la fonction *Read Input Register*.

3.2.3 Read Input Register

La fonction *Read Input Registers* permet de lire les dernières valeurs acquises par l'instrument (soit les mesures analogiques que celles numériques) et la date/heure du système. Les valeurs acquises peuvent être lues en format *floating point* en s'adressant à l'adresse 0 (0x0000) et en format binaire entier de l'adresse 1000 (0x03E8), tandis que la date/heure est disponible à l'adresse 2000 (0x07D0).

L'adresse à programmer pour chaque demande est le suivant:

Adresse (Hex)	Numéro de registres	Signification
Valeurs float IEEE754		
0x0000	2	Valeur de la mesure 1
0x0002	2	Valeur de la mesure 2
...
0x00C4	2	Valeur de la mesure 99
Valeurs entières (WORD)		
0x03E8	1	Valeur de la mesure 1
0x03E9	1	Valeur de la mesure 2
...
0x044B	1	Valeur de la mesure 99
Date/heure du système (yy MM dd hh mm ss)		
0x07D0	1	yy MM
0x07D1	1	dd hh
0x07D2	1	mm ss

Valeurs exprimées en format à virgule flottante

Toute mesure transmise avec virgule flottante est composée par 4 octets ou bien par 2 registres Modbus. Ils sont exprimés sur la base du format floating point indiqué par la norme IEEE754.

À travers le programme 3DOM on peut choisir le format des données entre *Big Endian* (réglage par défaut) et *Little Endian* (option *Inversione valore a virgola mobile/Inversion valeur à virgule flottante* disponible dans la fenêtre de réglage des paramètres de communication du port série 2). Le stockage de la donnée dans le premier cas commence par l'octet le plus significatif (MSB – Most Significant Byte), tandis que dans le deuxième cas il commence par l'octet le moins significatif (LSB – Least Significant Byte). La valeur 11,0 est donc stockée, à partir de l'adresse 0x20 selon l'exemple:

			Adresse de mémoire(Hex)	Valeur en format Big Endian (Hex)	Valeur en format Little Endian (Hex)
			...		
Valeur 11,0	Registre 1	Byte 1	0x20 (Hi)	00	30
		Byte 2	0x20 (Lo)	00	41
	Registre 2	Byte 1	0x21 (Hi)	41	00
		Byte 2	0x21 (Lo)	30	00
			...		

Puisque le cadre/frame est composé par 255 caractères, on peut obtenir 60 mesures maximum pour chaque message de demande. Pour avoir toutes les 99 mesures potentiellement disponibles dans l'enregistreur de données, il faudra donc effectuer deux demandes distinctes.

La valeur -999999 (0xF02374C9), sauf indication différente (chap. 3.2.6), correspond à *misura in errore/mesure en erreur*.

Demande

Code fonction	1 byte	0x04
Adresse de départ	2 byte	de 0x0000 à 0x00C4
Numéro de registres	2 byte	de 2 à 198 (max 120)

Réponse

Code fonction	1 byte	0x04
Numéro d'octets	1 byte	2 x N*
Valeur	2 byte x N*	

*N = numéro de mesures.

Erreur

Code fonction	1 byte	0x84
Code exception	1 byte	01 o 02 o 03 o 04

Pour plus d'informations sur *Codice eccezione*, faire référence au chap. 3.3.

Zone données

<i>No. Mesure</i>	1	2	3	4	5	6	...	99
<i>Adresse (hex)</i>	0x00	0x02	0x04	0x06	0x08	0x0A	...	0x0C4

Exemple: demande de lecture des valeurs, en format float, des mesures 3 et 4 de l'instrument ayant ID égal à 01:

<i>Demande</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	04
Adresse de départ (Hi)	00
Adresse de départ (Lo)	04
Numéro de sorties (Hi)	00
Numéro de sorties (Lo)	04
CRC16 (Hi)	B0
CRC16 (Lo)	08

La première mesure à lire est la numéro 3 et par conséquent, comme indiqué dans la *Zone données/ Area dati*, l'adresse de départ qu'on doit régler est 0x04 tandis que le numéro des sorties (registres) est 0x04 (2 mesures pour 2 registres).

Réponse à la demande avec les valeurs 99.0 pour la mesure 3 (valeur sortie 1) et 101.0 pour la mesure 4 (valeur sortie 2):

<i>Réponse</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	04
Numéro d'octets (bytes)	08
Valeur sortie 1 (byte 1)	00
Valeur sortie 1 (byte 2)	00
Valeur sortie 1 (byte 3)	42
Valeur sortie 1 (byte 4)	C6
Valeur sortie 2 (byte 1)	00
Valeur sortie 2 (byte 2)	00
Valeur sortie 2 (byte 3)	42
Valeur sortie 2 (byte 4)	C4
CRC16 (Hi)	13
CRC16 (Lo)	C9

Valeurs exprimés en format binaire entier

En format binaire, chaque mesure est de type entier, elle est donc composée de 2 octets, c'est à dire d'un registre Modbus. Le format des données est *Little Endian*. Le stockage de la donnée commence par l'octet moins significatif (LSB – Least Significant Byte). La valeur 1149 est ensuite stockée, à partir de l'adresse 0x03E8 selon l'exemple:

			Adresse de mémoire(Hex)	Valeur en format Little Endian (Hex)
Valeur 1149	Registre 1		...	
		Byte 1	0x03E8 (Hi)	05
		Byte 2	0x03E8 (Lo)	45

Pour ne pas perdre la partie décimale de la mesure on peut appliquer un offset et un gain à chaque valeur lue, en réglant respectivement *Costante matematica/ Constante mathématique* 1 et *Costante matematica 2* des grandeurs standard par moyen du programme 3DOM. De cette façon, ayant par exemple réglé un offset égal à 0 et un gain égal à 100, la valeur lue 11,49 deviendrait $(11,49 + 0) * 100 = 1149$. Pour obtenir la valeur correcte, la mesure reçue devra être divisée par le gain et éventuellement soustraite de l'offset.

Contrairement au format float, avec le format entier on peut obtenir toutes les 99 mesures potentiellement gérables par l'enregistreur de données à travers une demande unique.

La valeur -1, sauf indication différente (chap. 3.2.6), correspond à *misura in errore / mesure en erreur*.

Demande

<i>Code fonction</i>	1 byte	0x04
<i>Adresse de départ</i>	2 byte	de 0x03E8 à 0x044B
<i>Numéro de registres</i>	2 byte	de 1 à 99

Réponse

<i>Code fonction</i>	1 byte	0x04
<i>Numéro d'octets</i>	1 byte	2 x N*
<i>Valeur</i>	2 byte x N*	

*N = numéro de mesures.

Erreur

<i>Code fonction</i>	1 byte	0x84
<i>Code exception</i>	1 byte	01 ou 02 ou 03 ou 04

Pour plus d'informations sur *Codice eccezione (Code exception)* faire référence au chap. 3.3.

Zone données

<i>No. Mesure</i>	1	2	3	4	...	99
<i>Adresse (hex)</i>	0x03E8	0x03E9	0x03EA	0x03EB	...	0x044B

Exemple: demande de lecture des valeurs, en format entier, de la mesure 3 de l'instrument ayant ID égal à 01:

<i>Demande</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	04
Adresse de départ (Hi)	03
Adresse de départ (Lo)	EA
Numéro de sorties (Hi)	00
Numéro de sorties (Lo)	01
CRC16 (Hi)	A5
CRC16 (Lo)	BA

La mesure à lire est la numéro 3 et par conséquent, comme indiqué dans la *Zone données/ Area dati*, l'adresse de départ qu'on devra régler est 0x03EA tandis que le numéro de sorties (registres) est 0x01 (1 mesure pour 1 registre).

Réponse à demande avec la valeur 1343:

<i>Réponse</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	04
Numéro d'octets	02
Valeur sortie 1 (byte 1)	05
Valeur sortie 1 (byte 2)	3F
CRC16 (Hi)	FB
CRC16 (Lo)	04

Date/Heure de l'instrument

Demande

<i>Code fonction</i>	1 byte	0x04
<i>Adresse de départ</i>	2 byte	da 0x07D0 a 0x07D2
<i>Numéro de registres</i>	2 byte	da 1 a 3

Réponse

<i>Code fonction</i>	1 byte	0x04
<i>Numéro d'octets</i>	1 byte	2 x N*
<i>Valeur</i>	2 byte x N*	

*N = numéro de registres.

La séquence des champs date/heure à partir de l'adresse 0x07D0 est: yy MM dd hh mm ss.

Erreur

Code fonction	1 byte	0x84
Code exception	1 byte	01 ou 02 ou 03 ou 04

Pour plus d'informations sur *Codice eccezione* (*Code exception*) faire référence au chap. 3.3.

Exemple: demande de lecture de la date/heure de l'instrument ayant ID égal à 01:

Demande	
Nom du champ	(Hex)
Adresse dispositif	01
Code fonction	04
Adresse de départ (Hi)	07
Adresse de départ (Lo)	D0
Numéro de sorties (Hi)	00
Numéro de sorties (Lo)	03
CRC16 (Hi)	B0
CRC16 (Lo)	86

Réponse à la demande avec la date heure égale à 10/06/08 10:40:03:

Réponse	
Nom du champ	(Hex)
Adresse dispositif	01
Code fonction	04
Numéro de byte	06
Valeur sortie 1 (byte 1)	0A
Valeur sortie 1 (byte 2)	06
Valeur sortie 2 (byte 1)	08
Valeur sortie 2 (byte 2)	0A
Valeur sortie 3 (byte 1)	28
Valeur sortie 3 (byte 2)	03
CRC16 (Hi)	94
CRC16 (Lo)	5A

3.2.4 Write Single Coil

Utiliser la fonction *Write Single Coil* pour régler l'état des actionneurs de l'instrument (sorties numériques).

La valeur *0x0000* règle la sortie de l'actionneur sur *0* tandis que *0xFF00* la règle sur *1*. Normalement l'état *0* indique actionneur éteint et *1* actionneur allumé. Si au contraire la logique de fonctionnement des actionneurs en sécurité est réglée dans l'instrument, les états sont invertis : *0* indique actionneur allumé et *1* actionneur éteint (pour plus d'informations faire référence au manuel de l'instrument utilisé).

Demande

<i>Code fonction</i>	1 byte	0x05
<i>Adresse de départ</i>	2 byte	de 0x0000 à 0x0007
<i>Valeur</i>	2 byte	0x0000 ou 0xFF00

Réponse

<i>Code fonction</i>	1 byte	0x01
<i>Adresse de départ</i>	2 byte	de 0x0000 à 0x0007
<i>Valeur</i>	2 byte	0x0000 ou 0xFF00

Erreur

<i>Code fonction</i>	1 byte	0x85
<i>Code exception</i>	1 byte	01 ou 02 ou 03 ou 04

Pour plus d'informations sur *Codice eccezione* (*Code exception*) faire référence au chap. 3.3.

Exemple: demande de réglage sur 0 de l'état de l'actionneur numéro 3 de l'instrument ayant ID égal à 01:

<i>Demande</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	05
Adresse de départ (Hi)	00
Adresse de départ (Lo)	02
Valeur (Hi)	00
Valeur (Lo)	00
CRC16 (Hi)	6C
CRC16 (Lo)	0A

<i>Réponse</i>	
<i>Adresse dispositif</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	05
Adresse de départ (Hi)	00
Adresse de départ (Lo)	02
Valeur (Hi)	00
Valeur (Lo)	00
CRC16 (Hi)	6C
CRC16 (Lo)	0A

3.2.5 Write Multiple Coils

Utiliser la fonction *Write Multiple Coils* pour mettre à zéro des erreurs de fonctionnement éventuelles relevées avec la fonction *Read Coils*. La commande agit sur tous les bits ensemble; il n'est pas possible de mettre à zéro une erreur ou une partie d'une erreur. La commande, comme décrit ensuite, met à zéro toutes les erreurs survenues.

Demande

<i>Code fonction</i>	1 byte	0x0F
<i>Adresse de départ</i>	2 byte	0x0000
<i>Numéro de coil</i>	2 byte	32
<i>Numéro de byte</i>	1 byte	4
<i>Valeur</i>	4 byte	0x0000

Réponse

<i>Code fonction</i>	1 byte	0x0F
<i>Adresse de départ</i>	2 byte	0x0000
<i>Numéro de coil</i>	2 byte	32

Erreur

<i>Code fonction</i>	1 byte	0x8F
<i>Code exception</i>	1 byte	01 ou 02 ou 03 ou 04

Pour plus d'informations sur *Codice eccezione* (*Code exception*) faire référence au chap. 3.3.

Exemple: demande de mise à zéro des erreurs de fonctionnement de l'instrument ayant ID égal à 01:

<i>Demande</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	0F
Adresse de départ (Hi)	00
Adresse de départ (Lo)	00
Numéro de coil (Hi)	00
Numéro de coil (Lo)	20
Numéro de byte	04
Valeur 1 (Hi)	00
Valeur 1 (Lo)	00
Valeur 2 (Hi)	00
Valeur 2 (Lo)	00
CRC16 (Hi)	C4
CRC16 (Lo)	88

<i>Réponse</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01

Code fonction	0F
Adresse de départ (Hi)	00
Adresse de départ (Lo)	00
Numéro di coil (Hi)	00
Numéro di coil (Lo)	20
CRC16 (Hi)	21
CRC16 (Lo)	79

3.2.6 Write Multiple Registers

Utiliser la fonction *Write Multiple Registers* pour régler la date/heure de l'instrument et quelques paramètres de configuration des mesures.

L'adresse à utiliser pour chaque demande est la suivante:

<i>Adresse (Hex)</i>	<i>Numéro de registres</i>	<i>Signification</i>
<i>Date/heure du système (yy mm dd hh mm ss)</i>		
0x07D0	3	yy mm dd hh mm ss
<i>Configuration mesures</i>		
0x07DA	5	Valeur assumée pour la mesure en erreur par type décimal binaire, floating point et activation de la mesure (jusqu'à 32 max)

Les paramètres de configuration des mesures qui peuvent être modifiés sont:

- Valeur à assigner à la mesure en condition d'erreur pour les mesures transmises en format WORD (2 byte = 1 registre);
- Valeur à assigner à la mesure en condition d'erreur pour les mesures transmises en format float (4 byte = 2 registres); ATTENTION: utiliser le format *Little Endian*;
- Activation de la mesure (un bit pour chaque état d'activation de la mesure pour un maximum de 32 mesures; 4 bytes = 2 registres); la mesure non activée sera considérée en erreur.

Ces paramètres sont stockés de manière définitive jusqu'à leur prochaine reprogrammation ou jusqu'à la reconfiguration de l'instrument de la part de 3DOM. Dans ce cas, ils sont réglés avec des valeurs par défaut: -1 pour les valeurs décimales binaires -999999 pour les valeurs floating point, toutes les mesures sont habilitées.

L'écriture doit être effectuée entièrement sur tous les 5 registres.

Exemple 1: demande de modification de la date/heure de l'instrument avec la valeur 09/06/10 16:03:05 (format yy/mm/dd hh:mm:ss):

<i>Demande</i>	
<i>Nom du champ</i>	<i>(Hex)</i>
Adresse dispositif	01
Code fonction	10
Adresse de départ (Hi)	07
Adresse de départ (Lo)	D0
Numéro de registres (Hi)	00
Numéro de registres (Lo)	03
Numéro d'octets	06

Valeur 1 (Hi)	0A
Valeur 1 (Lo)	06
Valeur 2 (Hi)	09
Valeur 2 (Lo)	10
Valeur 3 (Hi)	03
Valeur 3 (Lo)	05
CRC16 (Hi)	B2
CRC16 (Lo)	5D

Réponse	
Nom du champ	(Hex)
Adresse dispositif	01
Code fonction	10
Adresse de départ (Hi)	07
Adresse de départ (Lo)	D0
Numéro de coil (Hi)	00
Numéro de coil (Lo)	03
CRC16 (Hi)	80
CRC16 (Lo)	85

Exemple 2: demande de modification des paramètres des mesures avec les valeurs suivantes: - 12345 (0xCFC7) pour une valeur en erreur de type décimal binaire, -12345678 (0x4E613CCB) pour une valeur en erreur de type float et désactivation de toutes les mesures à l'exception des premières 3:

Demande	
Nom du champ	(Hex)
Adresse dispositif	01
Code fonction	10
Adresse de départ (Hi)	07
Adresse de départ (Lo)	DA
Numéro de registres (Hi)	00
Numéro de registres (Lo)	05
Numéro d'octets	0A
Valeur 1 (Hi)	C7
Valeur 1 (Lo)	CF
Valeur 2 (Hi)	4E
Valeur 2 (Lo)	61
Valeur 3 (Hi)	3C
Valeur 3 (Lo)	CB
Valeur 4 (Hi)	07
Valeur 4 (Lo)	00
Valeur 5 (Hi)	00
Valeur 5 (Lo)	00
CRC16 (Hi)	6C
CRC16 (Lo)	11

Réponse	
Nom du champ	(Hex)
Adresse dispositif	01
Code fonction	10

Adresse de départ (Hi)	07
Adresse de départ (Lo)	DA
Numéro de coil (Hi)	00
Numéro de coil (Lo)	05
CRC16 (Hi)	20
CRC16 (Lo)	85

3.2.7 Read Device Identification

Utiliser la fonction *Read Device Identification* pour obtenir des informations sur l'instrument, comme : le nom de la société de production, type, code, numéro de série et version de l'instrument.

Demande

<i>Code fonction</i>	1 byte	0x2B
<i>Type MEI</i>	1 byte	0x0E
<i>Code Id lecture dispositif</i>	1 byte	01
<i>Id objet</i>	1 byte	0x00

Réponse

<i>Code fonction</i>	1 byte	0x2B
<i>Type MEI</i>	1 byte	0x0E
<i>Code Id lecture dispositif</i>	1 byte	01
<i>Niveau de conformité</i>	1 byte	0x01
<i>Id objet suivant</i>	1 byte	Numéro Id de l'objet
<i>Numéro des objets</i>	1 byte	3
<i>Id objet 1</i>	1 byte	0
<i>Longueur objet 1</i>	1 byte	26
<i>Valeur objet 1</i>	Longueur objet 1	"LSI-Lastem - Milan (Italy)"
<i>Id objet 2</i>	1 byte	1
<i>Longueur objet 2</i>	1 byte	31
<i>Valeur objet 2</i>	Longueur objet 2	Type, code, numéro de série d'usine et nom défini par l'utilisateur
<i>Id objet 3</i>	1 byte	2
<i>Longueur objet 3</i>	1 byte	8
<i>Valeur objet 3</i>	Longueur objet 3	Version de l'instrument

Erreur

<i>Code fonction</i>	1 byte	Code fonction + 0x80
<i>Code exception</i>	1 byte	01 ou 02 ou 03 ou 04

Pour plus d'informations sur *Codice eccezione (Code exception)* faire référence au chap. 3.3.

Exemple: ci-dessous, un exemple avec E-Log type 305, numéro de série 08030284/08030284, version du programme 2.08.01 avec ID réglé sur 01:

<i>Demande</i>	
<i>Nom du champ</i>	<i>Valeur</i>
Adresse dispositif	01
Code fonction	2B
Type MEI	0E
Code Id lecture dispositif	01
Id objet	00
CRC16 (Hi)	70
CRC16 (Lo)	77

<i>Réponse</i>	
<i>Nom du champ</i>	<i>Valeur</i>
Adresse dispositif	01
Code fonction	2B
Type MEI	0E
Code Id lecture dispositif	01
Niveau de conformité	01
Id objet suivant	00
Numéro d'objets	03
Id objet 1	00
Longueur objet 1	1A
Valeur objet 1*	“LSI-Lastem - Milan (Italy)”
Id objet 2	01
Longueur objet 2	21
Valeur objet 2*	“ELog-305; Serial08030284/08030284”
Id objet 3	02
Longueur objet 3	08
Valeur objet 3*	“02.08.01”
CRC16 (Hi)	9A
CRC16 (Lo)	6B

*Valeurs hexadécimales pour:

- Valeur objet 1: [4C][53][49][2D][4C][61][73][74][65][6D][20][2D][20][4D][69][6C][61][6E][20][28][49][74][61][6C][79][29][45][4C][6F][67][2D][33][30][35][3B][20][53][65][72][69][61][6C]
- Valeur objet 2: [30][38][30][33][30][32][38][34][2F][30][38][30][33][30][32][38][34]
- Valeur objet 3: [30][32][2E][30][38][2E][30][31]

3.3 Codes d'exception

Les codes d'exception sont transmis lorsque la commande envoyée au serveur, bien que en forme correcte, ne peut pas être effectuée. Les codes d'exception rendus sont les suivants:

<i>Code</i>	<i>Nom</i>	<i>Signification</i>
01	ILLEGAL FUNCTION	Le code de fonction ne correspond pas à une fonction supportée par le dispositif slave/serveur
02	ILLEGAL DATA ADDRESS	Le registre spécifié a une adresse non valide
03	ILLEGAL DATA VALUE	La valeur à assigner n'est pas valide pour l'adresse spécifiée

LSI LASTEM Datalogger – Protocoles de communication

04	SLAVE DEVICE FAILURE	Une erreur a eu lieu pendant l'exécution de la commande requise
----	----------------------	---