



Manuel Utilisateur

NT 100-RE-CO / NT 50-CO-EN

Passerelle NetTAP : Modbus TCP / CANopen Maitre
Firmware spécifique HFNTA002.NXF(NT100) et HFNTA003.NXF(NT50)
permettant la lecture / écriture de SDO
et la commande NMT
via Modbus TCP

Langue: Français

www.hilscher.com

Liste des révisions

Rev	Date	Name	
1	15.11.2012	JP / SLO	Création
2	12.07.2013	JP	Ajout NT50

Although this software has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this software for any purpose not confirmed by us in writing. Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this software or its documentation shall be limited to cases of intent. We reserve the right to modify our products and their specifications at any time in as far as this contribute to technical progress. The version of the manual supplied with the software applies.

Sommaire

1. Objet.....	4
2. Introduction.....	4
3. Mise en œuvre de la solution	5
4. Principe	5
5. Table des registres Modbus TCP.....	6
6. Résumé des commandes disponibles.....	7
7. Commande UPLOAD.....	8
8. Commande DOWNLOAD	9
9. Commande SET CONFIG	10
10. Commande SEND NMT	11
11. Diagnostic CANopen.....	12
12. Contact.....	13

1. Objet

Ce document décrit la mise en œuvre de la passerelle :

Hilscher netTAP NT100-RE-CO ou NT50-CO-EN

avec le firmware spécifique Modbus TCP / CANopen Maître :

HFNTA002.NXF(NT100) ou HFNTA003.NXF(NT50)

Ce firmware permet les fonctions de lecture / écriture de SDO et la commande NMT CANopen via Modbus TCP

2. Introduction

Hilscher propose avec ses passerelles netTAP 100, netTAP 50 et netBRICK et les firmwares standard disponibles, environ 300, un éventail de solutions pour la conversion de protocoles particulièrement large.

Cependant, si les firmwares standard couvrent très bien les besoins pour les données échangées cycliquement, comme les PDO pour CANopen, les communications acycliques ne sont à ce jour pas supportées.

Ces échanges acycliques sont utilisés pour la configuration, le paramétrage et le diagnostic précis des équipements connectés.

En visant l'exhaustivité en matière de combinaisons protocolaires, il n'est pas aisé de concevoir un mécanisme générique susceptible de supporter l'ensemble des fonctions quel que soit le protocole. Les concepts sont bien similaires mais les disparités sont irréductibles, volume, type de données, adressage...

D'où la nécessité de concevoir spécifiquement un mécanisme permettant cet échange acyclique. C'est ce que nous avons réalisé avec ce firmware.

Le produit standard Hilscher NT100-RE-CO et NT50-CO-EN possède les interfaces physiques nécessaires et Hilscher France fournit le firmware spécifique. Les outils de configuration standard Hilscher sont utilisés.

3. Mise en œuvre de la solution

La passerelle est livrée en standard avec un DVD contenant la documentation, l'outil de configuration SYCON.net et les pilotes nécessaires.

Le firmware spécifique est fourni avec ce document par Hilscher France.

Le firmware spécifique est chargé dans la passerelle comme le serait un firmware standard, avec SYCON.net.

Un projet SYCON.net exemple est fourni : NT_OMBCOM-ACYC

La seule différence avec la configuration d'un firmware standard se trouve au niveau de la fonction bridge. Elle n'est pas utilisée et il est inutile de procéder au mapping des données.

Un projet exemple Unity Pro pour un M340 est également fourni.

4. Principe

Côté Modbus TCP, la passerelle dispose d'une zone de registres en Entrée (vue côté client) et d'une zone de Sorties.

De l'offset 0 à l'offset 59 dans la zone des Sorties (vue côté client), on a disposé les registres qui permettent de gérer l'émission des requêtes.

De l'offset 0 à l'offset 59 dans la zone des Entrées (vue côté client), on a disposé les registres qui permettent de connaître l'état des requêtes.

Dans ces zones, on dispose également de deux registres permettant de spécifier la taille des données cycliques échangées.

On retrouvera ces données cycliques dans les registres 60 et suivants.

5. Table des registres Modbus TCP

Client Output				Client Input			
Registre Nbr	Registre	Min	Max	Registre Nbr	Registre	Min	Max
0	Cmd	0	40	0	Status	0	41
1	Noeud	0	127	1	Noeud	0	127
2	Index	0	65535	2	Index	0	65535
3	SubIndex	0	255	3	SubIndex	0	255
4	Data Size	0	4	4	Data Size	0	4
5	Data 0	0	65535	5	Data 0	0	65535
6	Data 1	0	65535	6	Data 1	0	65535
7	Data 2	0	65535	7	Data 2	0	65535
8	Data 3	0	65535	8	Data 3	0	65535
9	Dummy	0	65535	9	Error Low	0	65535
10	Dummy	0	65535	10	Error High	0	65535
11	Dummy	0	65535	11	Abort Low	0	65535
12	Dummy	0	65535	12	Abort High	0	65535
13	NMT State	1	130	13	NMT State	1	130
14	CyclicInSize	0	2480	14	CyclicInSize	0	2480
15	CyclicOutSize	0	2480	15	CyclicOutSize	0	2480
16	Dummy	0	0	16	Dummy	0	0
17	Dummy	0	0	17	Dummy	0	0
18	Dummy	0	0	18	Dummy	0	0
19	Dummy	0	0	19	Dummy	0	0
20	Dummy	0	65535	20	Dummy	0	65535
21	Dummy	0	65535	21	Dummy	0	65535
22	Dummy	0	65535	22	Dummy	0	65535
23	Dummy	0	65535	23	Dummy	0	65535
24	Dummy	0	65535	24	Dummy	0	65535
25	Dummy	0	65535	25	Dummy	0	65535
26	Dummy	0	65535	26	Dummy	0	65535
27	Dummy	0	65535	27	Dummy	0	65535
28	Dummy	0	65535	28	Extended status	0	65535
59	Dummy	0	65535	59	Extended status	0	65535
60	Cyclic Data	0	65535	60	Cyclic Data	0	65535
2879	Cyclic Data	0	65535	2879	Cyclic Data	0	65535

6. Résumé des commandes disponibles

Le registre 0 dans la zone des Sorties permet de spécifier la commande. Avant d'indiquer la commande, renseigner les champs de paramètres nécessaires.

Command	
0	CMD_NONE
10	CMD_UPLOAD
20	CMD_DOWNLOAD
30	CMD_SET_CONFIG
40	CMD_SEND_NMT

Le registre 0 dans la zone des Entrées permet de connaître l'état de la commande en cours.

Status	
0	CMD_STAT_NONE
10	CMD_STAT_UPLOADING
11	CMD_STAT_UPLOAD_DONE
20	CMD_STAT_DOWNLOADING
21	CMD_STAT_DOWNLOAD_DONE
30	CMD_STAT_SETTING_CONFIG
31	CMD_STAT_SET_CONFIG_DONE
40	CMD_STAT_SEND_NMT
41	CMD_STAT_SEND_NMT_DONE

7. Commande **UPLOAD**

La commande **UPLOAD** permet d'effectuer un **SDO UPLOAD**, autrement dit, une lecture d'un **SDO**.

1/ Ecrire les registres suivant :

Nœud : Nœud de l'équipement

Index : Index de l'objet

SubIndex : Sous Index de l'objet

Data Size : Nombre de donnée

2/ Ecrire la commande **CMD_UPLOAD** dans le registre **Cmd** pour lancer la commande.

Une fois le registre écrit, la commande est exécutée et le registre **Status** permet de voir si l'action est en cours ou bien terminée.

La commande est finie lorsque que le **Status** indique la valeur **CMD_STAT_UPLOAD_DONE**. Les registres **Error** et **Abort** permettent de voir si une erreur est survenue durant la commande.

3/ Ecrire la commande **CMD_NONE** dans le registre **Cmd** pour terminer la commande.

8. Commande **DOWNLOAD**

La commande **DOWNLOAD** permet d'effectuer un **SDO DOWNLOAD**, autrement dit, une écriture d'un **SDO**.

1/ Ecrire les registres suivant :

Nœud : Nœud de l'équipement

Index : Index de l'objet

SubIndex : Sous Index de l'objet

Data Size : Nombre de donnée

Data 0 : Premier mot (Word) de donnée (octets de donnée 1 et 2)

Data 1 : Deuxième mot (Word) de donnée (octets de donnée 3 et 4)

Data 2 : Troisième mot (Word) de donnée (octets de donnée 5 et 6)

Data 3 : Quatrième mot (Word) de donnée (octets de donnée 7 et 8)

2/ Ecrire la commande **CMD_DOWNLOAD** dans le registre **Cmd** pour lancer la commande.

Une fois le registre écrit, la commande est exécutée et le registre **Status** permet de voir si l'action est en cours ou bien terminée.

La commande est finie lorsque que le **Status** indique la valeur **CMD_STAT_DOWNLOAD_DONE**. Les registres **Error** et **Abort** permettent de voir si une erreur est survenue durant la commande.

3/ Ecrire la commande **CMD_NONE** dans le registre **Cmd** pour terminer la commande.

9. Commande SET CONFIG

La commande SET CONFIG permet de configurer la taille des données cyclique à échanger.

1/ Ecrire les registres suivant :

CyclicInSize : Nombre de registres cycliques en entrée

CyclicOutSize : Nombre de registres cycliques en sortie

2/ Ecrire la commande CMD_SET_CONFIG dans le registre **Cmd** pour lancer la commande.

Une fois le registre écrit, la commande est exécutée et le registre **Status** permet de voir si l'action est en cours ou bien terminée.

La commande est finie lorsque que le **Status** indique la valeur CMD_STAT_SET_CONFIG_DONE. Le registre **Error** permet de voir si une erreur est survenue durant la commande.

3/ Ecrire la commande CMD_NONE dans le registre **Cmd** pour terminer la commande.

Les valeurs configurées sont visibles de la passerelle dans les registres **CyclicInSize** et **CyclicOutSize** dans la zone en Entrée. Ces valeurs sont sauvegardées et restaurées en cas de Reset de la passerelle.

10. Commande SEND NMT

La commande SEND NMT permet d'envoyer des commandes NMT.

1/ Ecrire les registres suivant :

NMTState : Etat NMT

Value	Symbolic Name	Meaning
1	CANOPEN_MASTER_NMT_COMMAND_START	Starts the node
2	CANOPEN_MASTER_NMT_COMMAND_STOP	Stops the node
128	CANOPEN_MASTER_NMT_COMMAND_ENTER_PREOPERATIONAL	Sets the node into the preoperational state
129	CANOPEN_MASTER_NMT_COMMAND_RESET_NODE	Resets the node
130	CANOPEN_MASTER_NMT_COMMAND_RESET_COMMUNICATION	Resets the communication to the node

Table 83: NMT States

2/ Ecrire la commande CMD_SEND_NMT dans le registre **Cmd** pour lancer la commande.

Une fois le registre écrit, la commande est exécutée et le registre **Status** permet de voir si l'action est en cours ou bien terminée.

La commande est finie lorsque que le **Status** indique la valeur CMD_STAT_SEND_NMT_DONE. Les registres **Error** et **Abort** permettent de voir si une erreur est survenue durant la commande.

3/ Ecrire la commande CMD_NONE dans le registre **Cmd** pour terminer la commande.

11. Diagnostic CANopen

Un diagnostic de la partie CANopen est effectué cycliquement, le résultat est disponible dans les registres **Extended status**.

Le détail du diagnostic est décrit ci-dessous :

Status Block for CANopen Master		
Type	Name	Description
unsigned char	bGlobalBits	Global error bits
unsigned char	bCanState	Main state of the CANopen Master system
unsigned char	bErrorNodeAddress	Unused
unsigned char	bErrorEvent	Unused
unsigned short	usBusErrorCount	Number of detected bus error limit oversteps
unsigned short	usBusOffCount	Number of CAN-chip reinitializations
unsigned short	usMsgTimeOut	Number of cancelled CAN messages, because of getting no message acknowledging partner
unsigned short	usRxOverFlow	Number of indicated Receive Message Overflow coming from the CAN chip
unsigned char[]	abGlobalError[4]	Reserved error variables
unsigned char[]	abListProjectedNodes[16]	List of projected nodes
unsigned char[]	abListActivatedNodes[16]	List of activated nodes
unsigned char[]	abListDiagnosticNodes[16]	List of diagnostic nodes

12. Contact

Headquarter

Germany

Hilscher Gesellschaft für
Systemautomation mbH
Rheinstrasse 15
65795 Hattersheim
Phone: +49 (0) 6190 9907-0
Fax: +49 (0) 6190 9907-50
E-Mail: <mailto:info@hilscher.com>

Support

Phone: +49 (0) 6190 9907-99
E-Mail: de.support@hilscher.com

Subsidiaries

China

Hilscher Ges.f.Systemaut. mbH
Shanghai Representative Office
200010 Shanghai
Phone: +86 (0) 21-6355-5161
E-Mail: info@hilscher.cn

Support

Phone: +86 (0) 21-6355-5161
E-Mail: cn.support@hilscher.com

France

Hilscher France S.a.r.l.
69500 Bron
Phone: +33 (0) 4 72 37 98 40
E-Mail: info@hilscher.fr

Support

Phone: +33 (0) 4 72 37 98 40
E-Mail: fr.support@hilscher.com

Italy

Hilscher Italia srl
20090 Vimodrone (MI)
Phone: +39 02 25007068
E-Mail: info@hilscher.it

Support

Phone: +39 / 02 25007068
E-Mail: it.support@hilscher.com

Japan

Hilscher Japan KK
Tokyo, 160-0022
Phone: +81 (0) 3-5362-0521
E-Mail: info@hilscher.jp

Support

Phone: +81 (0) 3-5362-0521
E-Mail: jp.support@hilscher.com

Switzerland

Hilscher Swiss GmbH
4500 Solothurn
Phone: +41 (0) 32 623 6633
E-Mail: info@hilscher.ch

Support

Phone: +49 (0) 6190 9907-99
E-Mail: ch.support@hilscher.com

USA

Hilscher North America, Inc.
Lisle, IL 60532
Phone: +1 630-505-5301
E-Mail: info@hilscher.us

Support

Phone: +1 630-505-5301
E-Mail: mailto:us.support@hilscher.com