

Description
EtherNet/IP
Interface process

Contenu

1.	Remarques préliminaires	3
2.	Remarques générales.....	3
3.	Réglages EtherNet/IP	3
3.1	Structure de l'assemblage en cas de segmentation désactivée	4
3.2	Segmentation.....	4
3.2.1	Exemple de segmentation 1	5
3.2.2	Exemple de segmentation 2	5
4.	Echange de données via EtherNet/IP.....	5

1. Remarques préliminaires

Ce document sert à expliquer une communication EtherNet/IP entre un capteur de vision ifm possédant une interface EtherNet/IP et un API compatible avec EtherNet/IP.

2. Remarques générales

Ethernet Industrial Protocol (EtherNet/IP) est un standard ouvert pour les réseaux industriels. EtherNet/IP sert à transmettre des données E/S cycliques ainsi que des données de paramètres acycliques. Ainsi, EtherNet/IP offre d'innombrables possibilités pour une communication de données efficace dans l'industrie.

EtherNet/IP enrichit Ethernet d'un protocole industriel moderne (CIP, Common Industrial Protocol) en tant que couche d'application pour les applications d'automatisation. Ainsi, Ethernet est parfaitement adapté aux systèmes de contrôle-commande industriels.

3. Réglages EtherNet/IP

En tant que EtherNet/IP Adapter Device, le capteur de reconnaissance d'objets supporte la communication avec un appareil configuré comme scanner EtherNet/IP. Normalement, il s'agit du système de commande (par exemple un API).

La communication s'effectue à l'aide de deux assemblages EtherNet/IP - l'un pour le transfert de données du système de commande vers le capteur ("Output Assembly Instance", adresse ID 100 / 0x64) et l'autre pour le transfert de données du capteur vers le système de commande ("Input Assembly Instance", adresse ID 101 / 0x65).

Les réglages des longueurs des assemblages dans le capteur et dans le système de commande doivent être identiques.

- "Longueur de l'assemblage récepteur" définit la longueur de "Output Assembly Instance" (ID 100)
- "Appliquer la segmentation" active la définition des différents paramètres "Input Assembly Instance" (ID 101).

3.1 Structure de l'assemblage en cas de segmentation désactivée

i	"Input Assembly Instance" (101) est de 450 bytes de long et segmenté en 3 sections :			
	Offset	Longueur	Contenu	
	Section 1	0	215 bytes	Réponse aux messages arrivants
	Section 2	215	215 bytes	Résultat en bytes des évaluations des codes ou des images
Section 3	430	20 bytes	Informations fixes sur les appareils et les résultats	

Le dernier byte de chaque segment sert d'octet de contrôle qui est incrémenté pendant le traitement par le capteur de reconnaissance d'objets. Si le contenu du code est identique, ces octets de contrôle servent à différencier les données d'entrée.

Test	Offset	Longueur	Désignation champ à cocher
Octet de contrôle section 1	214	1 byte	Réponse
Octet de contrôle section 2	429	1 byte	Evaluation
Octet de contrôle section 3	449	1 byte	Informations sur l'appareil

3.2 Segmentation

Dans "Output Assembly Instance", les messages sont toujours écrits à partir de l'adresse 0 ; ici, seulement la longueur peut être définie. Elle doit au moins correspondre à la longueur du message le plus long possible (450 bytes maximum).

Pour économiser de la mémoire, "Input Assembly Instance" peut être segmentée. Ainsi, seulement les données qui sont vraiment nécessaires pour l'application sont transférées. Pour chaque segment, un "offset" et la "longueur" requise peuvent être sélectionnés dans la liste de sélection des segments.

La division "Input Assembly Instance" prédéfinie peut être reconfigurée à l'aide du tableau de segmentation. Le tableau de segmentation indique une nouvelle séquence des bytes pour "Input Assembly Instance". Un segment se définit par son index, un nombre de bytes (longueur du segment) et une adresse d'octet de la "Input Assembly Instance" prédéfinie (Segment Offset).

Ainsi, "Input Assembly Instance" est divisée de nouveau. L'index du segment détermine la séquence d'attribution. Le nombre de bytes attribués se définit par la longueur du segment tandis que l'offset du segment fait référence à l'adresse de "Input Assembly Instance" prédéfinie d'où sont extraits les bytes.

Le dernier byte de chaque segment peut être activé comme octet de contrôle. Celui-ci est incrémenté par le capteur de reconnaissance d'objets pendant le traitement. Si le contenu du code est identique, ces octets de contrôle servent à différencier les données d'entrée. En cliquant sur le champ respectif vous pouvez activer ou désactiver l'octet de contrôle.

Attention ! Pour une segmentation correcte, tous les bytes de la "Input Assembly Instance" doivent être indiqués de manière univoque.

3.2.1 Exemple de segmentation 1

Tableau de segmentation

Index	Offset	Longueur
1	0	450 bytes

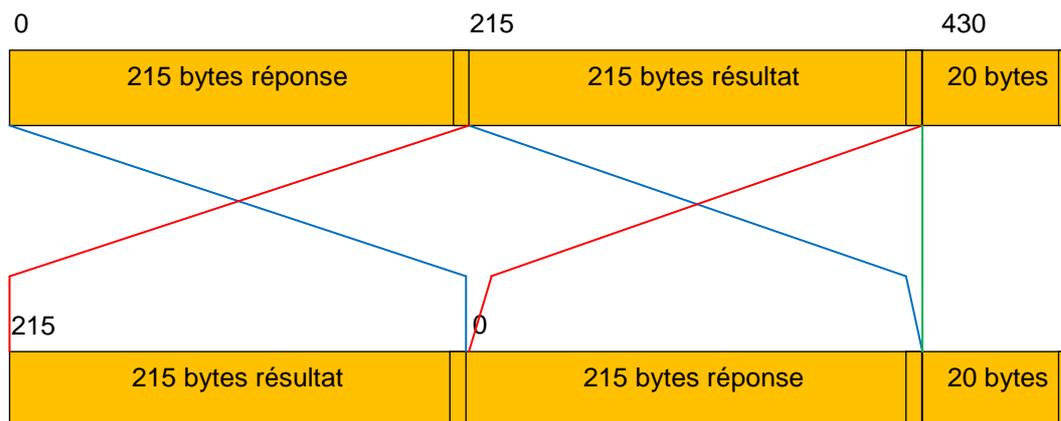
Explication :

Tableau de segmentation par défaut. Prend 450 bytes (tous !) de la "Input Assembly Instance" prédéfinie et les place à l'adresse 0. Pour cette raison, le tableau de segmentation n'a pas vraiment d'effet sur "Input Assembly Instance".

3.2.2 Exemple de segmentation 2

Tableau de segmentation

Index	Offset	Longueur
1	215	215 bytes
2	0	215 bytes
3	430	20 bytes



Explication :

Les sections 1 et 2 de la "Input Assembly Instance" prédéfinie sont échangées : D'abord, 215 bytes de la "Input Assembly Instance" prédéfinie sont déplacés à partir de l'adresse d'octet 215 et après 215 bytes à partir de l'adresse 0 et puis 20 bytes à partir de l'adresse 430 de la "Input Assembly Instance" prédéfinie.

4. Echange de données via EtherNet/IP

L'échange de données entre un capteur compatible avec EtherNet/IP et un API s'effectue de manière cyclique.

C'est-à-dire, les données sauvegardées dans le capteur dans la section Output-Assembly (ID 100) sont accueillies par l'API raccordé pendant chaque cycle et sauvegardées dans la section de données définie dans l'API.

Lorsque les données dans le capteur changent, elles sont adoptées dans la section de données définie pendant le prochain cycle de l'API et sont disponibles jusqu'à ce que le capteur réécrive sa section Output-Assembly.