

Interface de commande intuitive
Automate Modicon M340
Description de la bibliothèque

DTE102

Contenu

Contenu	2
1 Aperçu	3
2 Exclusion de la responsabilité	3
3 Régler les paramètres de connexion dans le boîtier de contrôle	3
4 Projeter la connexion dans Unity Pro	4
4.1 Ajouter le boîtier de contrôle dans le catalogue matériel d'Unity Pro	4
4.2 Ajouter le boîtier de contrôle dans le projet logiciel	6
4.3 Régler les paramètres de connexion dans le module de communication	8
4.4 Configurer les voies du boîtier de contrôle	11
5 Travailler avec la bibliothèque	12
5.1 Intégrer la bibliothèque dans Unity Pro	12
5.2 Contenu de la bibliothèque	12
5.3 Utiliser les blocs fonctionnels dans le programme d'application	13
6 Les blocs fonctionnels de la bibliothèque en détail	13
6.1 Le bloc fonctionnel DTE102_INACTIVE	13
6.2 Le bloc fonctionnel DTE102_INPUT	14
6.3 Le bloc fonctionnel DTE102_OUTPUT	16
6.4 Le bloc fonctionnel DTE102_RWH_RW	18
6.4.1 Structure de la communication	20
6.4.2 Processus du traitement de commandes	21
6.4.3 Lecture de l'UID du tag	22
6.4.4 Lecture d'une zone de la mémoire USER du tag	23
6.4.5 Ecriture sur une zone de la mémoire USER du tag	23
6.4.6 Indications sur le paramètre de sortie <i>owStatus</i>	24

1 Aperçu

La bibliothèque DTE102 pour Unity Pro contient un package de codes pour l'automate Modicon M340 de Schneider Electric pour simplifier l'échange de données avec le boîtier de contrôle DTE102 d'ifm electronic dans le programme d'application de l'API. Pour chaque module supporté du boîtier de contrôle un bloc fonctionnel (DFB) est mis à disposition.

Ce document explique la configuration de la connexion entre l'automate et le boîtier de contrôle et décrit l'intégration et l'utilisation de la bibliothèque dans le projet Unity Pro.

2 Exclusion de la responsabilité

Le package de codes de la bibliothèque est une version de démonstration. Il est uniquement destiné à servir d'exemple pour l'utilisateur. Toute utilisation de ce package de codes pour la commande machine s'effectue sous la responsabilité du développeur !

Les auteurs et titulaires d'un droit de ce package de codes excluent toute responsabilité pour le bon fonctionnement ou la compatibilité de cette version de démonstration.

Pour des raisons de droits de propriété intellectuelle seule la distribution gratuite de ce package de codes est autorisée.

3 Régler les paramètres de connexion dans le boîtier de contrôle

L'interface web intégré dans l'appareil est utilisé pour le réglage des paramètres de connexion dans le boîtier de contrôle. Ceci est possible avec tout navigateur web. Le boîtier de contrôle est livré avec l'adresse IP 192.168.0.79.

- Raccorder le PC et le boîtier de contrôle dans un réseau
- Démarrer le navigateur web sur le PC et saisir l'adresse IP du boîtier de contrôle
- A la page *Home* cliquer sur le lien [EDS.ZIP](#) pour télécharger une archive avec le fichier EDS pour l'appareil.

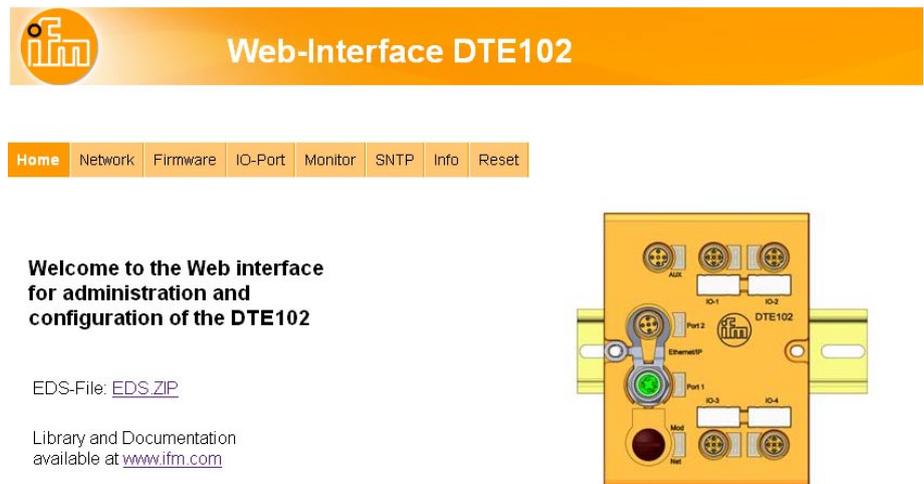
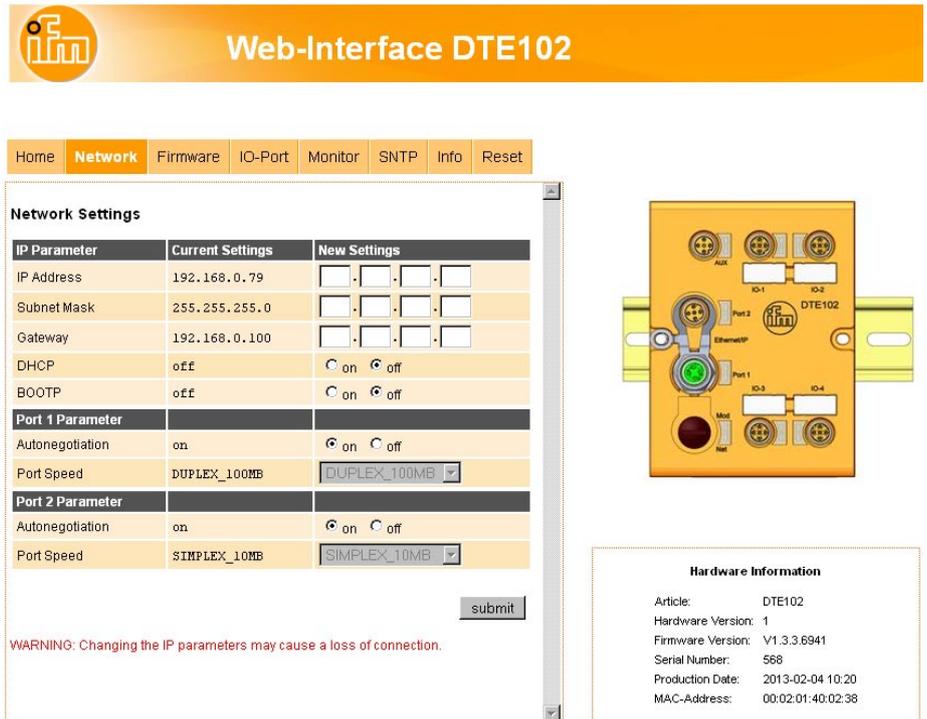


Figure 1

- A la page *Network* les paramètres du réseau peuvent être réglés
- Cliquer sur [submit] pour appliquer les paramètres du réseau dans le boîtier de contrôle



The screenshot shows the 'Web-Interface DTE102' with a navigation menu (Home, Network, Firmware, IO-Port, Monitor, SNTP, Info, Reset) and a 'Network Settings' form. The form includes sections for IP Parameters, Port 1, and Port 2, with a 'submit' button and a warning message.

IP Parameter	Current Settings	New Settings
IP Address	192.168.0.79	<input type="text"/> . <input type="text"/> . <input type="text"/> . <input type="text"/>
Subnet Mask	255.255.255.0	<input type="text"/> . <input type="text"/> . <input type="text"/> . <input type="text"/>
Gateway	192.168.0.100	<input type="text"/> . <input type="text"/> . <input type="text"/> . <input type="text"/>
DHCP	off	<input type="radio"/> on <input checked="" type="radio"/> off
BOOTP	off	<input type="radio"/> on <input checked="" type="radio"/> off
Port 1 Parameter		
Autonegotiation	on	<input checked="" type="radio"/> on <input type="radio"/> off
Port Speed	DUPLEX_100MB	<input type="text" value="DUPLEX_100MB"/>
Port 2 Parameter		
Autonegotiation	on	<input checked="" type="radio"/> on <input type="radio"/> off
Port Speed	SIMPLEX_10MB	<input type="text" value="SIMPLEX_10MB"/>

submit

WARNING: Changing the IP parameters may cause a loss of connection.

Hardware Information

Article: DTE102
 Hardware Version: 1
 Firmware Version: V1.3.3.6941
 Serial Number: 568
 Production Date: 2013-02-04 10:20
 MAC-Address: 00:02:01:40:02:38

Figure 2

4 Projeter la connexion dans Unity Pro

La communication entre l'automate et le boîtier de contrôle s'effectue via Ethernet/IP grâce à un module de communication NOC0401 dans l'automate M340. Les blocs fonctionnels de la bibliothèque DTE102 utilisent le service cyclique "Implicit Messaging" pour l'échange de données avec le boîtier de contrôle. Pour ce faire, une projection de la connexion dans Unity Pro est impérative.

Chaque boîtier de contrôle DTE102 utilise 80 octets dans la zone d'entrées et 80 octets dans la zone de sorties du module de communication NOC0401. Ces zones sont à réserver lors de la configuration des zones d'adressage du module de communication.

4.1 Ajouter le boîtier de contrôle dans le catalogue matériel d'Unity Pro

Pour intégrer le boîtier de contrôle dans le catalogue matériel le fichier EDS du boîtier de contrôle est nécessaire. Ce fichier EDS peut être téléchargé via l'interface web du boîtier de contrôle comme archive (voir le chapitre 3). Après le décompactage de cette archive le fichier EDS peut être intégré dans le catalogue matériel.

- Dans le navigateur de DTM cliquer sur le module de communication par le bouton droit de la souris
- Sélectionner via le menu contextuel *Menu Equipement* → *Fonctions supplémentaires* → *Ajouter un fichier EDS à la bibliothèque*

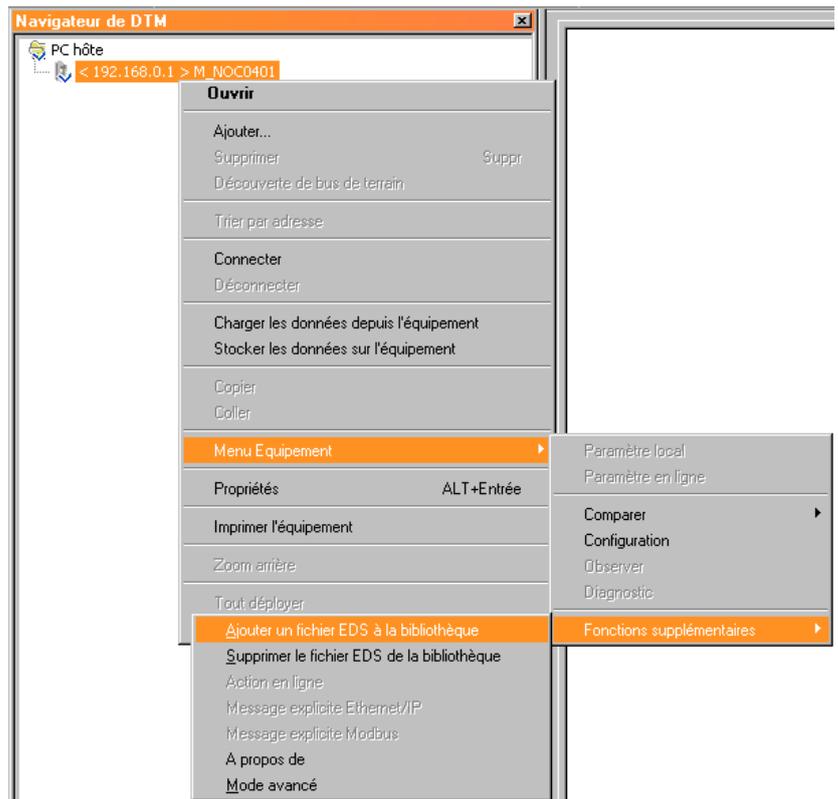


Figure 3

- A l'aide de l'assistant qui s'ouvre ajouter le fichier EDS du boîtier de contrôle à la bibliothèque

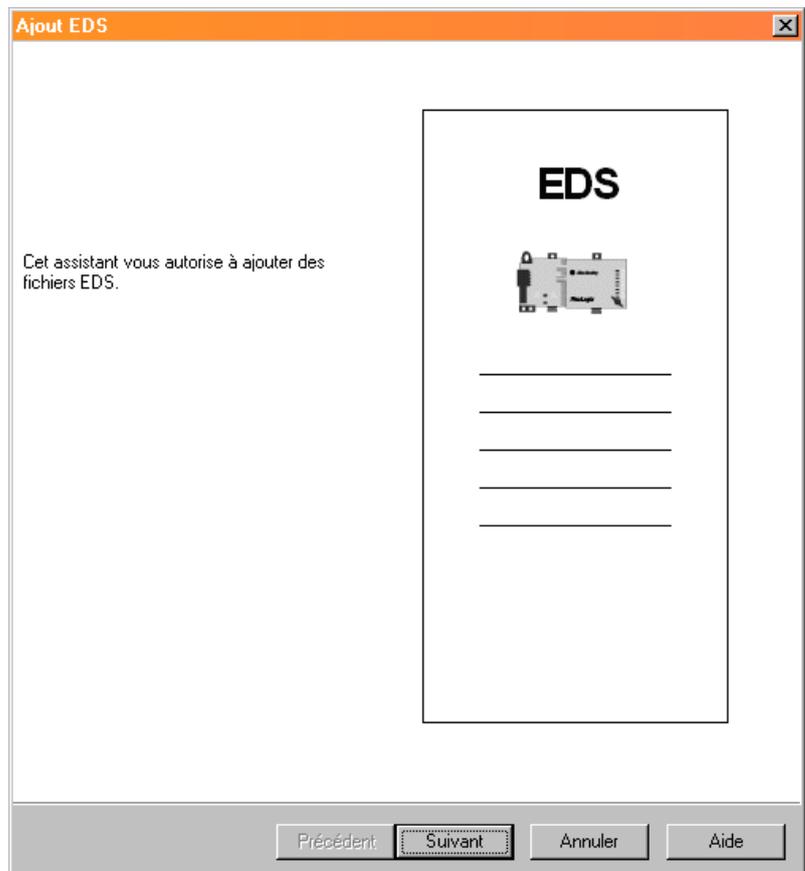


Figure 4

- Ouvrir le catalogue matériel dans Unity Pro

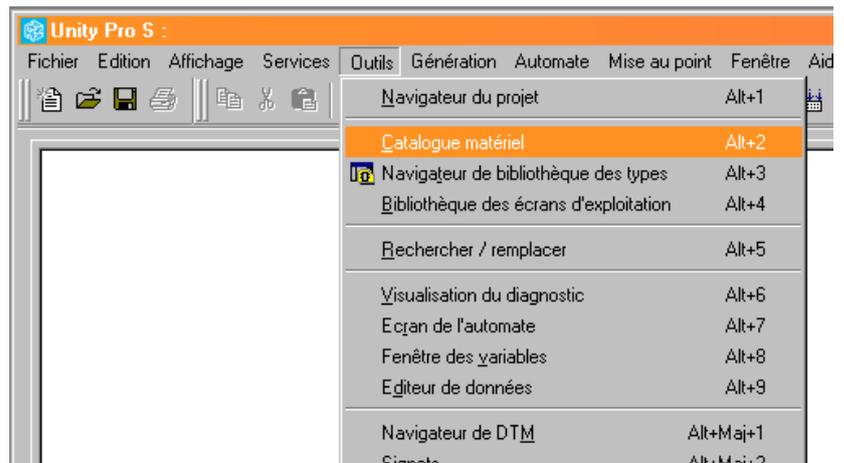


Figure 5

- Dans le catalogue matériel passer à l'onglet *Catalogue DTM*
- Cliquer sur [Mettre à jour] pour mettre le catalogue à jour

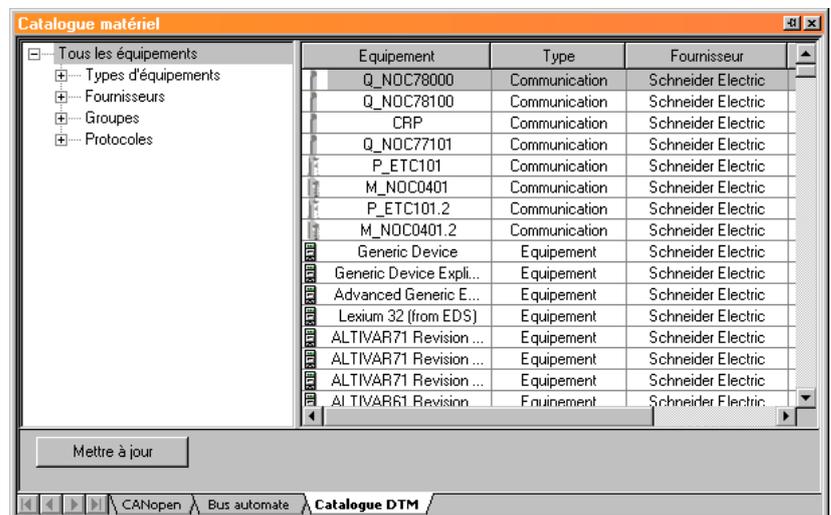


Figure 6

4.2 Ajouter le boîtier de contrôle dans le projet logiciel

- Dans le navigateur de DTM cliquer sur le module de communication via le bouton droit de la souris
- Sélectionner via le menu contextuel *Ajouter...*

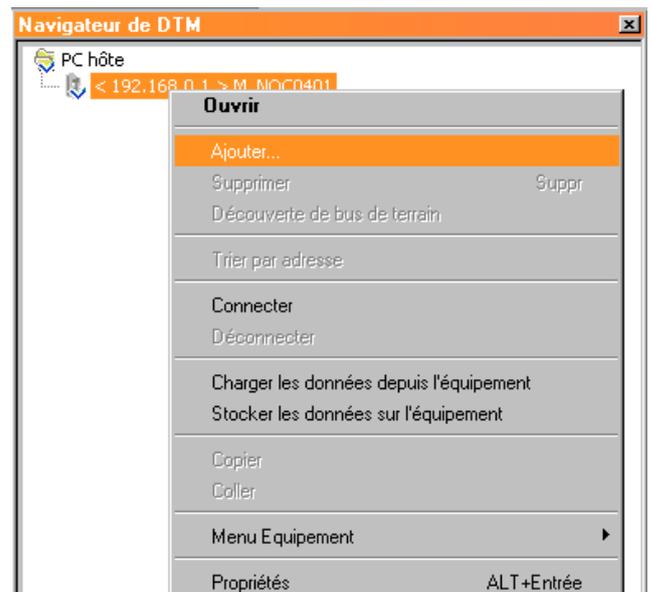


Figure 7

- Sélectionner *DTE102XX (from EDS)* de la liste des équipements
- Cliquer sur [Ajouter DTM] pour ouvrir la fenêtre *Propriétés de l'équipement*

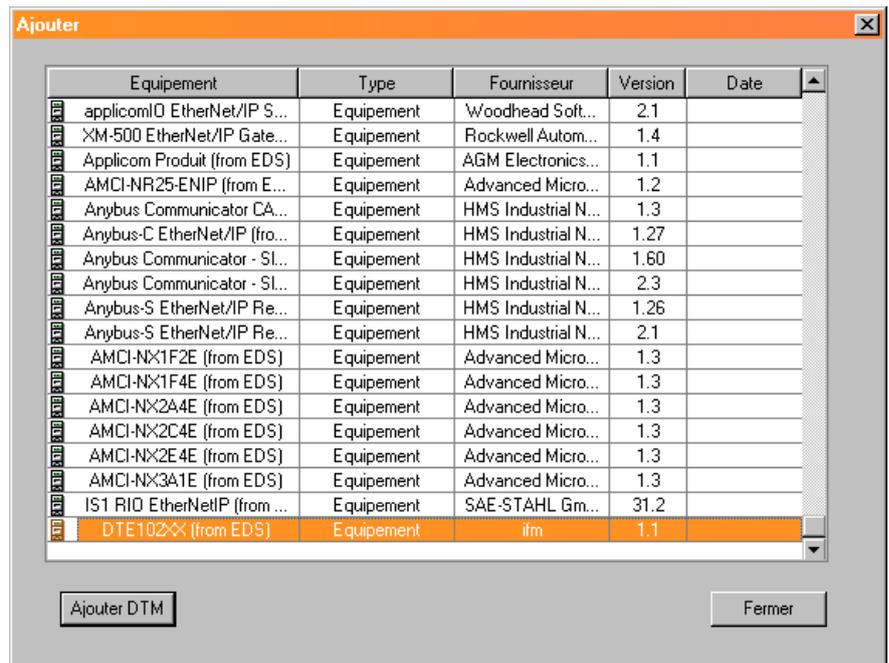


Figure 8

- Dans l'onglet *Général* donner un nom d'alias unique à l'équipement

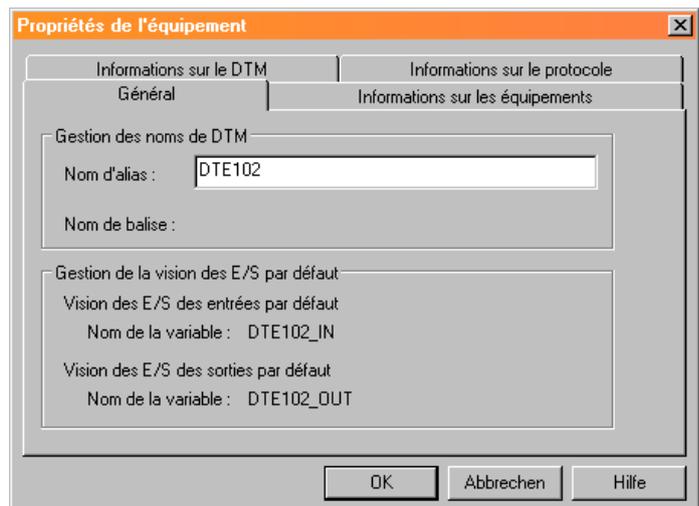


Figure 9

- Dans l'onglet *Informations sur le protocole* sélectionner le protocole *CIP (EtherNet/IP)*
- Quitter la fenêtre *Propriétés de l'équipement* en cliquant sur [OK] pour appliquer les sélections

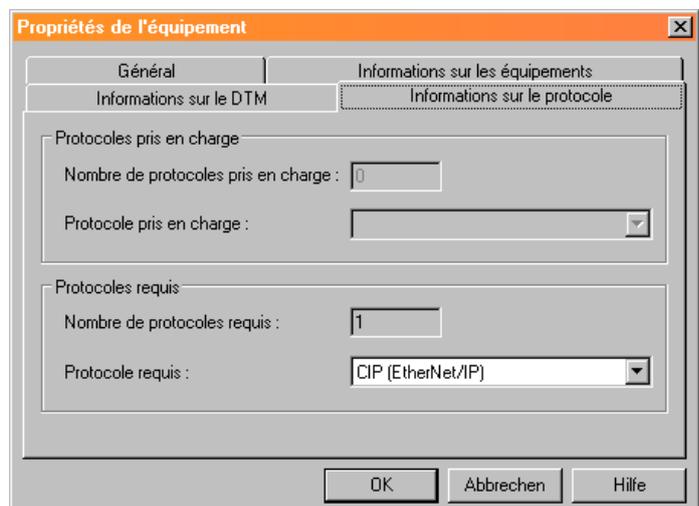


Figure 10

4.3 Régler les paramètres de connexion dans le module de communication

- Dans le navigateur de DTM cliquer sur le module de communication via le bouton droit de la souris
- Sélectionner via le menu contextuel *Ouvrir* pour ouvrir la fenêtre de configuration du module de communication



Figure 11

- Dans la structure sur le côté gauche de la fenêtre élargir le point *Liste des équipements* et sélectionner le boîtier de contrôle
- Sélectionner l'onglet *Propriétés*
- Dans la zone *Gestion des items* sélectionner *Manuel* pour le *Mode d'importation*

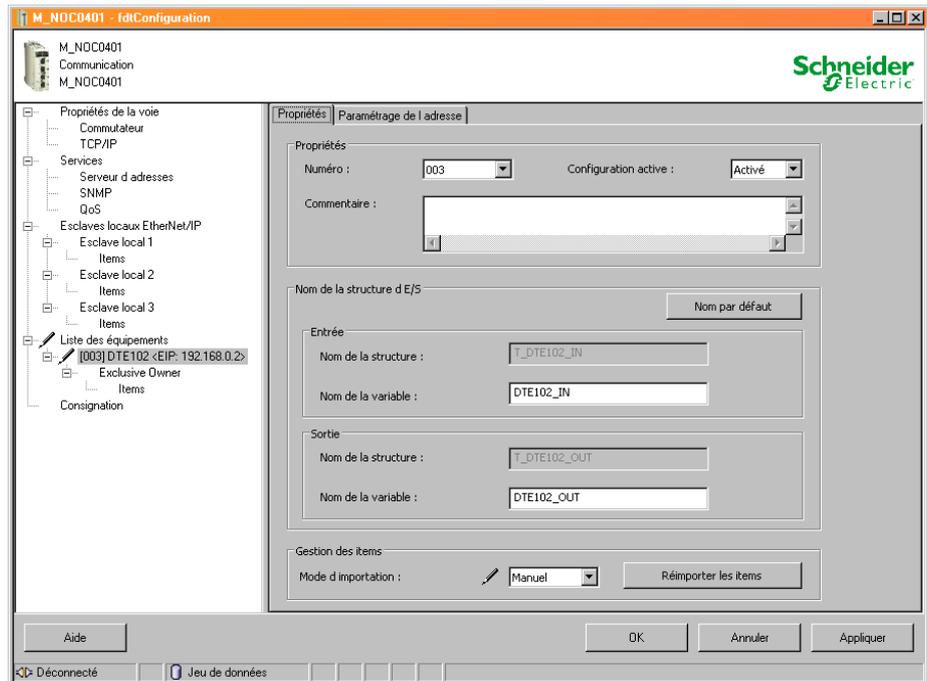


Figure 12

- Passer à l'onglet *Paramétrage de l'adresse*
- Dans la zone *Modifier l'adresse* saisir l'adresse IP réglée dans le boîtier de contrôle

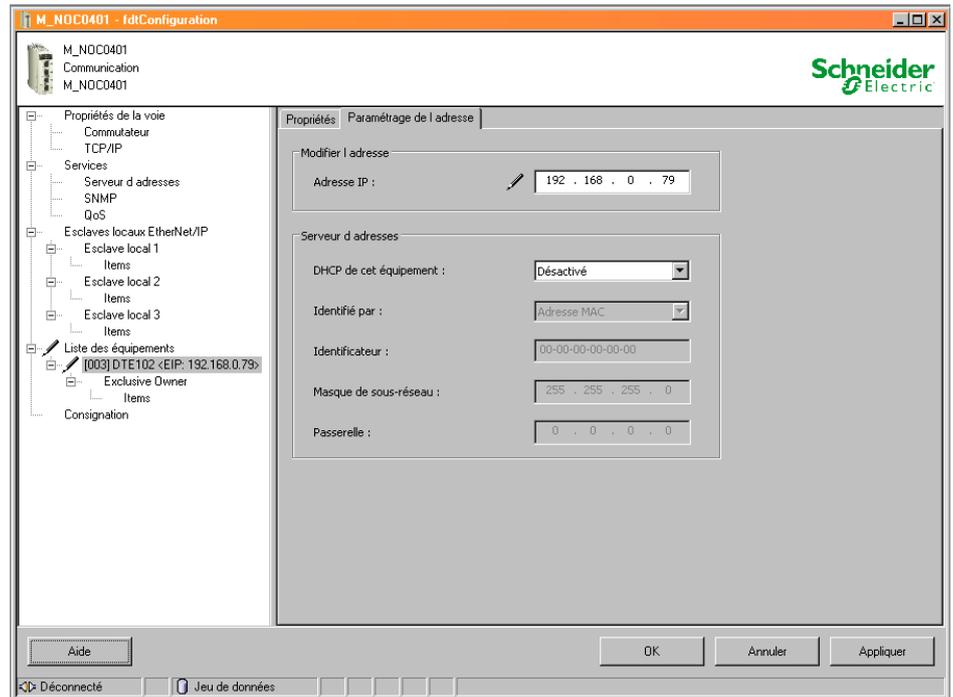


Figure 13

- Dans la structure sur le côté gauche de la fenêtre sélectionner le point *Exclusive Owner*
- Régler le paramètre *Request Packet Interval (RPI) >= 10 ms*

La valeur *Bit de connexion* indique le numéro du bit qui montre l'état de la connexion Ethernet/IP vers l'appareil dans les HEALTH_BITS. Les HEALTH_Bits font partie de l'array d'octets d'état dans la zone d'entrées du module de communication. Par exemple le bit de connexion est = 0 dans la **Figure 14**. C'est-à-dire que le bit HEALTH_BITS_IN[0].0 montre si la connexion Ethernet/IP vers le boîtier de contrôle est établie.

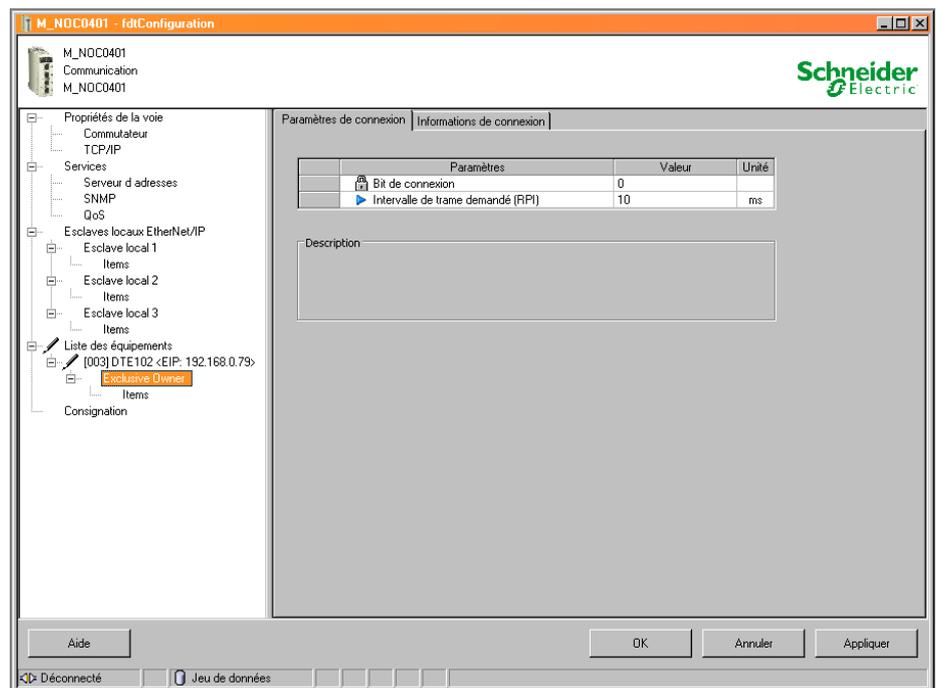


Figure 14

- Dans la structure sur le côté gauche de la fenêtre sélectionner le point *Items*

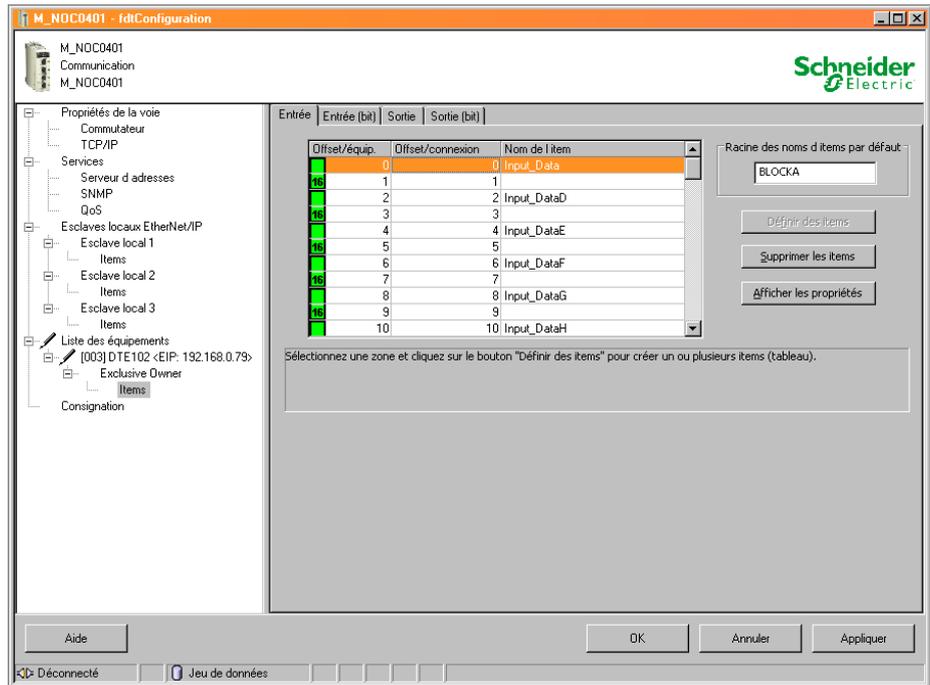


Figure 15

- Dans les onglets *Entrée*, *Entrée (bit)*, *Sortie* et *Sortie (bit)* effacer tous les items prédéfinis
- Appliquer les sélections en cliquant sur [OK] et fermer la fenêtre

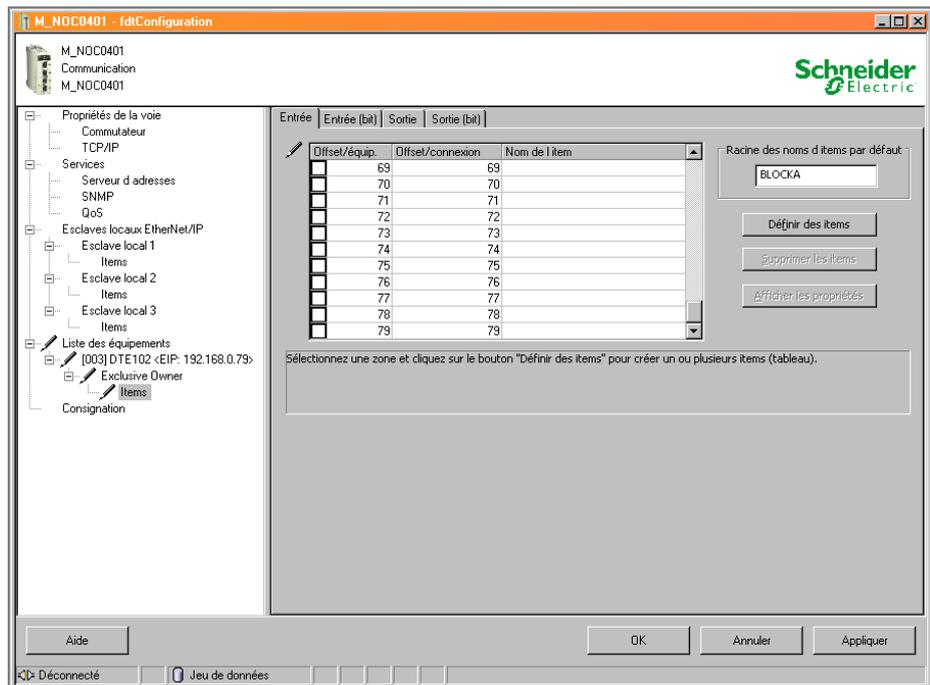


Figure 16

4.4 Configurer les voies du boîtier de contrôle

- Dans le navigateur de DTM cliquer sur le symbole du boîtier de contrôle via le bouton droit de la souris
- Sélectionner via le menu contextuel *Ouvrir* pour ouvrir la fenêtre de configuration du boîtier de contrôle

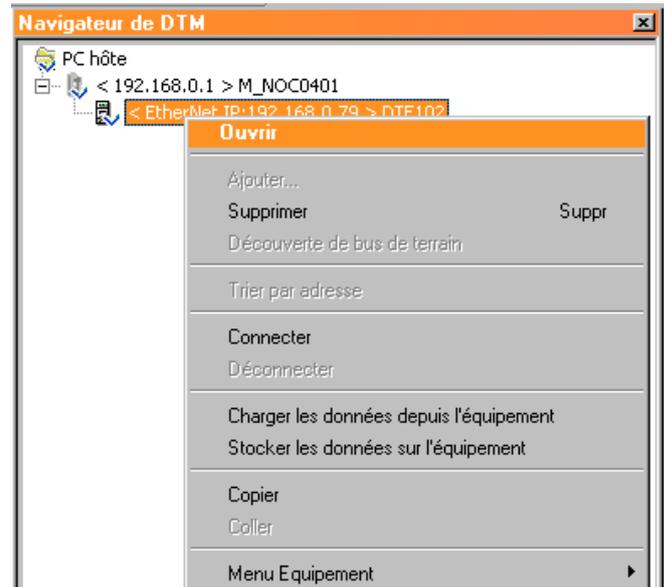


Figure 17

- Dans la structure cliquer sur *Exclusive Owner*
- Passer à l'onglet *Paramètres de configuration*
- Configurer toutes les voies du boîtier de contrôle en fonction de l'utilisation prévue.

Les paramètres sont décrits dans le manuel du boîtier de contrôle.

- Cliquer sur [OK] pour appliquer les données

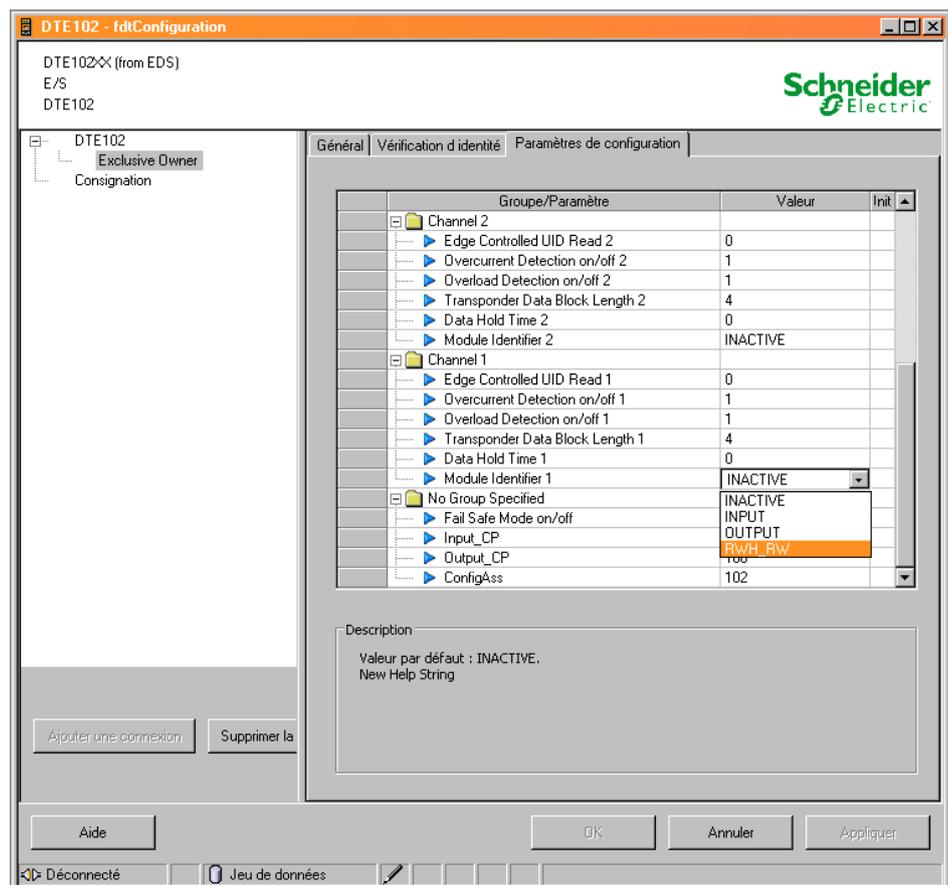


Figure 18

5 Travailler avec la bibliothèque

5.1 Intégrer la bibliothèque dans Unity Pro

La bibliothèque DTE102 est fournie comme fichier d'archive. Après le décompactage de cette archive la bibliothèque peut être intégrée dans la bibliothèque de types d'Unity Pro grâce à l'outil pour la mise à jour de la bibliothèque de types du progiciel Unity.

- Démarrer l'outil pour la mise à jour de la bibliothèque de types
- Comme source sélectionner le répertoire avec la bibliothèque
- Cliquer sur [Installer famille]
- Après l'installation cliquer sur [Quitter] pour quitter l'outil.

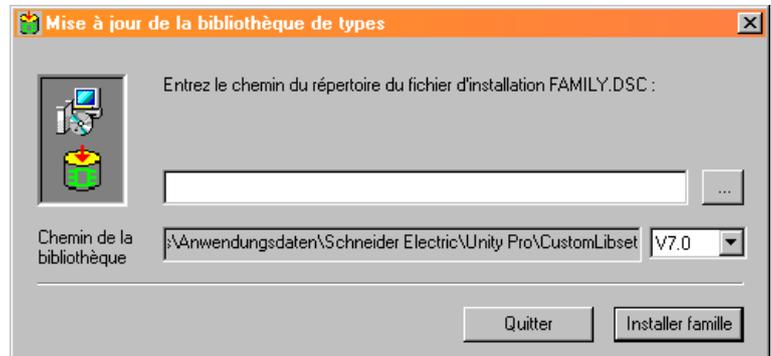


Figure 19

Dans la bibliothèque *ifm electronic* les blocs sont rangés dans la famille *DTE102*.

<Libset V7.0>	
Base Lib	This is the main manufacturer library. It contains the main IEC families and s...
Communication	This library is used for exchanging data between devices connected by a b...
CONT_CTL	This library is for projecting process-engineering servo-loops. It contains Con...
Custom Lib	This Library is used to store Customer DFBs and DDTs.
Diagnostics	This library is used to investigate the control program for misbehaviours. It c...
I/O Management	This library contains EFBs, which are required for using I/O modules. It cont...
ifm electronic	This library contains DFBs and DDTs for devices made by ifm electronic.
DTE102	This library contains DFBs for module types of the evaluation unit DTE102
MotionFunctionBlock	This library contains Motion Function Blocks.
Motion	This Library contains the Axis Control, CAM control, Lexium and MMF Start ...
Obsolete Lib	This library is used for compatibility with PL7 or Concept applications. It cont...
Safety	This library contains safety EF/EFBs for the usage in safety applications.
System	This library contains Events, SFC Management, SysClock Families.
UnityLL984	LL984 Librarv for Unity Pro. Contains FFBs for exclusive usage in LL984 edit...

Figure 20

5.2 Contenu de la bibliothèque

La bibliothèque DTE102 contient un bloc fonctionnel spécifique à l'utilisateur (DFB) pour chaque module supporté du boîtier de contrôle.

Nom	Type	Description
DTE102_INACTIVE	DFB	Bloc fonctionnel pour une voie du boîtier de contrôle DTE102 avec identificateur de module = INACTIVE (pour plus de détails voir le chapitre 6.1)
DTE102_INPUT	DFB	Bloc fonctionnel pour une voie du boîtier de contrôle DTE102 avec l'identificateur de module = INPUT (pour plus de détails voir le chapitre 6.2)
DTE102_OUTPUT	DFB	Bloc fonctionnel pour une voie du boîtier de contrôle DTE102 avec l'identificateur de module = OUTPUT (pour plus de détails voir le chapitre 6.3)
DTE102_RWH_RW	DFB	Bloc fonctionnel pour une voie du boîtier de contrôle DTE102 avec l'identificateur de module = RWH_RW (pour plus de détails voir le chapitre 6.4)

Tableau 1

5.3 Utiliser les blocs fonctionnels dans le programme d'application

Pour chaque voie un bloc fonctionnel peut être utilisé dans le programme d'application. Ce bloc doit correspondre au module sélectionné de la voie (voir Tableau 1

). La sélection correcte n'est pas vérifiée dans les blocs fonctionnels.

Une seule instance d'un bloc fonctionnel par voie d'un boîtier de contrôle est permise. Toutes les instances utilisées doivent être appelées une fois par cycle dans le programme cyclique de l'API.



Pour l'utilisation des blocs fonctionnels, des arrays dynamiques (ANY_ARRAY_XXX) doivent être déclarées comme variables ainsi que l'extraction de bits depuis des octets dans le projet Unity. Ceci est défini dans les propriétés du projet.

6 Les blocs fonctionnels de la bibliothèque en détail

6.1 Le bloc fonctionnel DTE102_INACTIVE

Ce bloc fonctionnel peut être utilisé pour une voie du boîtier de contrôle DTE102 paramétrée comme INACTIVE. Il offre la fonction suivante :

Lecture des données de diagnostic de la voie du boîtier de contrôle

Dans chaque cycle API le bloc fonctionnel extrait les données de la voie depuis les données d'entrée du boîtier de contrôle. Si un événement de diagnostic se produit sur la voie, les données de diagnostic sont lues automatiquement et indiquées sur les paramètres de sortie. La lecture des données de diagnostic fonctionne de manière asynchrone, c'est-à-dire qu'elle s'étend sur plusieurs cycles API.

La figure suivante montre un appel du bloc fonctionnel DTE102_INACTIVE.

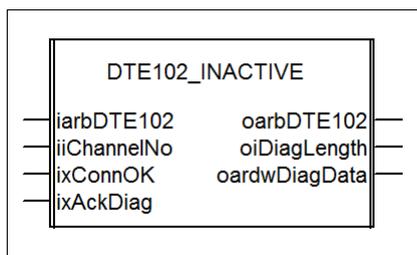


Figure 21

Description des paramètres :

Paramètres d'entrée :

Nom	Type	Description
iarbDTE102	ANY_ARRAY_BYTE	Champ de données des entrées du boîtier de contrôle
iiChannelNo	DI	Numéro de voie (1..4)
ixConnOK	BOOL	Connexion Ethernet/IP vers le boîtier de contrôle établie
ixAckDiag	BOOL	Effacement des données de diagnostic indiquées

Tableau 2

Paramètres de sortie :

Nom	Type	Description
oarbDTE102	ANY_ARRAY_BYTE	Champ de données des sorties vers le boîtier de contrôle
oiDiagLength	INT	Nombre des saisies dans les données de diagnostic
oardwDiagData	ARRAY[0..3] OF DWORD	Array avec données de diagnostic (pour les codes d'erreur voir le manuel DTE102)

Tableau 3

Les données d'entrée du boîtier de contrôle sont mises à disposition pour le bloc fonctionnel sur le paramètre d'entrée *iarbDTE102*. Le paramètre de sortie *oarbDTE102* renvoie aux données de sortie qui sont transférées au boîtier de contrôle.

Le paramètre d'entrée *iiChannelNo* détermine quelle voie du boîtier de contrôle doit être traitée.

Le paramètre d'entrée *ixConnOK* indique au bloc fonctionnel que la connexion Ethernet/IP vers le boîtier de contrôle est établie. De préférence, ce paramètre d'entrée est à lier au bit de connexion correspondant (**Figure 14**) de la zone d'entrées du module de communication.

Tant que le paramètre d'entrée *ixConnOK* est = FALSE, les données de diagnostic indiquées sur les paramètres de sortie sont initialisées avec 0. Si le signal sur le paramètre d'entrée *ixConnOK* est = TRUE, le bloc fonctionnel commence à travailler.

Si le boîtier de contrôle signale une erreur sur la voie paramétrée, le bloc fonctionnel ajoute automatiquement les signaux de commande pour la lecture des données de diagnostic dans les données de sortie vers le boîtier de contrôle. Les données de diagnostic reçues sont indiquées dans le paramètre de sortie *oardwDiagData*. Les données de diagnostic indiquées jusqu'à présent sont entièrement effacées. Le paramètre de sortie *oiDiagLength* montre combien de jeux de données de diagnostic sont indiqués actuellement.

Les codes d'erreur dans les données de diagnostic sont décrits dans le manuel du boîtier de contrôle.

Un front montant sur le paramètre d'entrée *ixAckDiag* efface les données de diagnostic indiquées.

6.2 Le bloc fonctionnel DTE102_INPUT

Ce bloc fonctionnel peut être utilisé pour une voie du boîtier de contrôle DTE102 paramétrée comme INPUT.

Il offre les fonctions suivantes :

- Lecture du signal numérique sur l'entrée C/Qi

- Lecture du signal numérique sur l'entrée I/Q

- Surveillance de la charge sur la connexion L+ du port

- Lecture des données de diagnostic de la voie du boîtier de contrôle

Dans chaque cycle API le bloc fonctionnel extrait les données de la voie depuis les données d'entrée du boîtier de contrôle et met à disposition les données des entrées TOR de la voie sur ses paramètres de sortie. Si un événement de diagnostic se produit sur la voie, les données de diagnostic sont lues automatiquement et indiquées sur les paramètres de sortie. La lecture des données de diagnostic fonctionne de manière asynchrone et s'étend sur plusieurs cycles API. Ceci n'a aucune influence sur les fonctions cycliques du bloc. Si ce bloc fonctionnel est utilisé pour une voie qui n'est pas paramétrée comme une INPUT, il en résulte un mauvais fonctionnement du bloc.

La figure suivante montre un appel du bloc fonctionnel DTE102_INPUT.

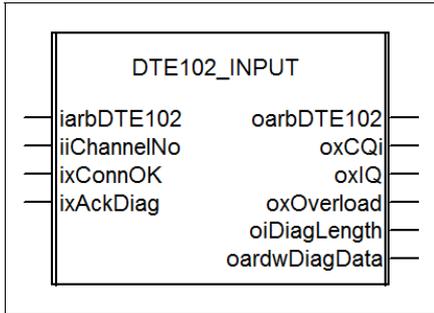


Figure 22

Description des paramètres :

Paramètres d'entrée :

Nom	Type	Description
iarbDTE102	ANY_ARRAY_BYTE	Champ de données des entrées du boîtier de contrôle
iiChannelNo	INT	Numéro de voie (1..4)
ixConnOK	BOOL	Connexion Ethernet/IP vers le boîtier de contrôle établie
ixAckDiag	BOOL	Effacement des données de diagnostic indiquées

Tableau 4

Paramètres de sortie :

Nom	Type	Description
oarbDTE102	ANY_ARRAY_BYTE	Champ de données des sorties vers le boîtier de contrôle
oxCQi	BOOL	Etat de l'entrée TOR C/Qi
oxIQ	BOOL	Etat de l'entrée TOR I/Q
oxOverload	BOOL	Surcharge de l'alimentation en tension via L+
oiDiagLength	INT	Nombre des saisies dans les données de diagnostic
oardwDiagData	ARRAY[0..3] OF DWORD	Array avec données de diagnostic (pour les codes d'erreur voir le manuel DTE102)

Tableau 5

Les données d'entrée du boîtier de contrôle sont mises à disposition au bloc fonctionnel sur le paramètre d'entrée *iarbDTE102*. Le paramètre de sortie *oarbDTE102* renvoie aux données de sortie qui sont transférées au boîtier de contrôle.

Le paramètre d'entrée *iiChannelNo* détermine quelle voie du boîtier de contrôle doit être traitée.

Le paramètre d'entrée *ixConnOK* indique au bloc fonctionnel que la connexion Ethernet/IP vers le boîtier de contrôle est établie. De préférence, ce paramètre d'entrée est à lier au bit de connexion correspondant (**Figure 14**) de la zone d'entrées du module de communication.

Tant que le paramètre d'entrée *ixConnOK* est = FALSE, tous les paramètres de sortie sont initialisés avec 0.

Si le signal sur le paramètre d'entrée *ixConnOK* est = TRUE, le bloc fonctionnel commence à travailler.

Une voie du boîtier de contrôle paramétrée comme INPUT met à disposition 2 entrées TOR. L'état de ces entrées est indiqué sur les paramètres de sortie *oxCQi* et *oxIO*. Une surcharge de l'alimentation en tension via ces entrées est signalée sur le paramètre de sortie *oxOverload*.

Si le boîtier de contrôle signale une erreur sur la voie paramétrée, le bloc fonctionnel ajoute automatiquement les signaux de commande pour la lecture des données de diagnostic dans les données de sortie vers le boîtier de contrôle. Les données de diagnostic reçues sont indiquées dans le paramètre de sortie *oardwDiagData*. Les données de diagnostic indiquées jusqu'à présent sont entièrement effacées. Le paramètre de sortie *oiDiagLength* montre combien de jeux de données de diagnostic sont indiqués actuellement.

Les codes d'erreur dans les données de diagnostic sont décrits dans le manuel du boîtier de contrôle. Un front montant sur le paramètre d'entrée *ixAckDiag* efface les données de diagnostic indiquées.

6.3 Le bloc fonctionnel DTE102_OUTPUT

Ce bloc fonctionnel peut être utilisé pour une voie paramétrée du boîtier de contrôle DTE102 comme OUTPUT. Il offre les fonctions suivantes :

- Commande du signal numérique sur la sortie C/Qo
- Lecture du signal numérique sur l'entrée I/Q
- Activation de l'alimentation en courant augmentée de la sortie C/Qo (seulement disponible pour voies 3 et 4)
- Surveillance de la charge sur la connexion L+ du port
- Lecture des données de diagnostic de la voie du boîtier de contrôle

Dans chaque cycle API le bloc fonctionnel extrait les données de la voie depuis les données d'entrée du boîtier de contrôle et met à disposition les données de l'entrée TOR de la voie sur ses paramètres de sortie. Les signaux pour la sortie TOR de la voie sont ajoutés de manière cyclique dans les données de sortie vers le boîtier de contrôle.

Si un événement de diagnostic se produit sur la voie, les données de diagnostic sont lues automatiquement et indiquées sur les paramètres de sortie. La lecture des données de diagnostic fonctionne de manière asynchrone et s'étend sur plusieurs cycles API. Ceci n'a aucune influence sur les fonctions cycliques du bloc. Si ce bloc fonctionnel est utilisé pour une voie qui n'est pas paramétrée comme OUTPUT, il en résulte un mauvais fonctionnement du bloc.

La figure suivante montre un appel du bloc fonctionnel DTE102_OUTPUT.

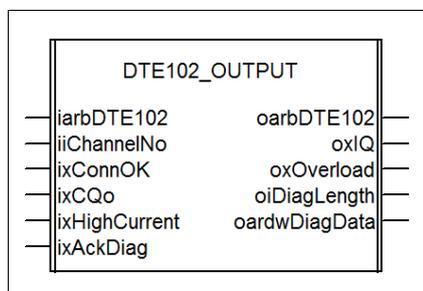


Figure 23

Description des paramètres :

Paramètres d'entrée :

Nom	Type	Description
iarbDTE102	ANY_ARRAY_BYTE	Champ de données des entrées du boîtier de contrôle
iiChannelNo	INT	Numéro de voie (1..4)
ixConnOK	BOOL	Connexion Ethernet/IP vers le boîtier de contrôle établie
ixCQo	BOOL	Signal pour la sortie C/Qo
ixHighCurrent	BOOL	Activation de l'alimentation en courant augmentée de la sortie C/Qo
ixAckDiag	BOOL	Effacement des données de diagnostic indiquées

Tableau 6

Paramètres de sortie :

Nom	Type	Description
oarbDTE102	ANY_ARRAY_BYTE	Champ de données des sorties vers le boîtier de contrôle
oxIQ	BOOL	Etat de l'entrée TOR I/Q
oxOverload	BOOL	Surcharge de l'alimentation en tension via L+
oiDiagLength	INT	Nombre des saisies dans les données de diagnostic
oardwDiagData	ARRAY[0..3] OF DWORD	Array avec données de diagnostic (pour les codes d'erreur voir le manuel DTE102)

Tableau 7

Les données d'entrée du boîtier de contrôle sont mises à disposition au bloc fonctionnel sur le paramètre d'entrée *iarbDTE102*. Le paramètre de sortie *oarbDTE102* renvoie aux données de sortie qui sont transférées au boîtier de contrôle.

Le paramètre d'entrée *iiChannelNo* détermine quelle voie du boîtier de contrôle doit être traitée.

Le paramètre d'entrée *ixConnOK* indique au bloc fonctionnel que la connexion Ethernet/IP vers le boîtier de contrôle est établie. De préférence, ce paramètre d'entrée est à lier au bit de connexion correspondant (**Figure 14**) de la zone d'entrées du module de communication.

Tant que le paramètre d'entrée *ixConnOK* est = FALSE, tous les paramètres de sortie sont initialisés avec 0. Si le signal sur le paramètre d'entrée *ixConnOK* est = TRUE, le bloc fonctionnel commence à travailler.

Une voie du boîtier de contrôle paramétrée comme OUTPUT met à disposition 1 sortie TOR et 1 entrée TOR. L'état de l'entrée est indiqué sur le paramètre de sortie *oxI/O*. Le signal sur le paramètre d'entrée *ixCQo* est ajouté dans les données de sortie vers le boîtier de contrôle et commande la sortie TOR de la voie. Le signal sur le paramètre d'entrée *ixHighCurrent* est seulement pris en considération pour les voies 3 et 4 et transféré au boîtier de contrôle. Une surcharge de l'alimentation en tension via l'entrée ou la sortie est signalée sur le paramètre de sortie *oxOverload*.

Si le boîtier de contrôle signale une erreur sur la voie paramétrée, le bloc fonctionnel ajoute automatiquement les signaux de commande pour la lecture des données de diagnostic dans les données de sortie vers le boîtier de contrôle. Les données de diagnostic reçues sont indiquées dans le paramètre de sortie *oardwDiagData*. Les données de diagnostic indiquées jusqu'à présent sont entièrement effacées. Le paramètre de sortie *oiDiagLength* montre combien de jeux de données de diagnostic sont indiqués actuellement.

Les codes d'erreur dans les données de diagnostic sont décrits dans le manuel du boîtier de contrôle.

Un front montant sur le paramètre d'entrée *ixAckDiag* efface les données de diagnostic indiquées.

6.4 Le bloc fonctionnel DTE102_RWH_RW

Ce bloc fonctionnel peut être utilisé pour une voie du boîtier de contrôle DTE102 paramétrée comme RWH_RW. A l'aide d'une antenne RFID raccordée, des données peuvent être lues depuis un tag ou mémorisées sur un tag.



Lorsque ce bloc fonctionnel est utilisé, le paramètre *Data Hold Time* doit être réglé = 0 pour le paramétrage de la voie (Figure 18).

Le bloc fonctionnel offre les fonctions suivantes :

- Détection de la présence d'un tag
- Lecture de l'UID d'un tag
- Lecture d'une zone de la mémoire USER depuis un tag
- Écriture sur une zone de la mémoire USER d'un tag
- Mise hors tension et sous tension de l'antenne
- Lecture des données de diagnostic de la voie du boîtier de contrôle

L'écriture de l'UID n'est pas possible. La tentative est finie avec un message d'erreur.

Si ce bloc fonctionnel est utilisé pour une voie qui n'est pas paramétrée comme RWH_RW, il en résulte un mauvais fonctionnement du bloc.

Le bloc fonctionnel extrait les données de la voie depuis les données d'entrée du boîtier de contrôle et les évalue. Des signaux de commande pour la lecture d'un tag ou l'écriture sur un tag sont ajoutés dans les données de sortie vers le boîtier de contrôle. Si un événement de diagnostic se produit sur la voie, les données de diagnostic sont lues automatiquement et indiquées sur les paramètres de sortie.

La communication doit s'effectuer selon une procédure définie. Ceci est montré à l'aide d'un diagramme dans la figure 24.

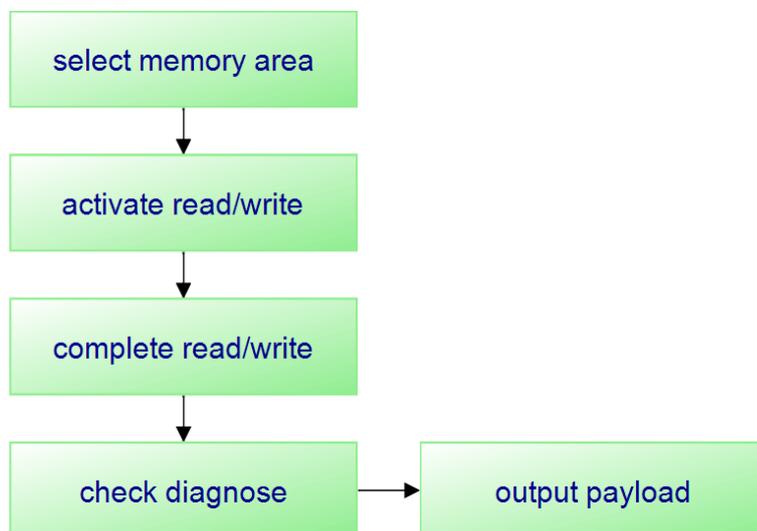


Figure 24

Pour rendre le traitement de la commande plus convivial, le bloc fonctionnel *DTE102_RWH_RW* a été créé. Ce bloc traite automatiquement le protocole de commande montré ci-dessus. La figure suivante montre un appel du bloc fonctionnel *DTE102_RWH_RW*.

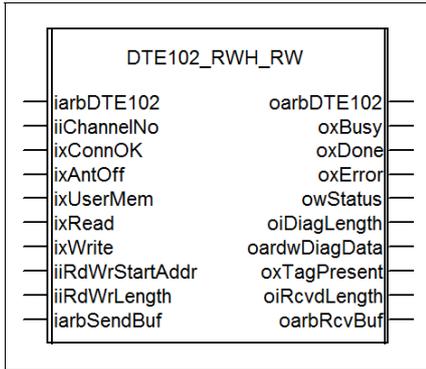


Figure 25

Description des paramètres :

Paramètres d'entrée :

Nom	Type	Description
iarbDTE102	ANY_ARRAY_BYTE	Champ de données des entrées du boîtier de contrôle
iiChannelNo	INT	Numéro de voie (voie 1..4)
ixConnOK	BOOL	Connexion Ethernet/IP vers le boîtier de contrôle établie
ixAntOff	BOOL	Désactivation de l'antenne
ixUserMem	BOOL	Accès à la mémoire USER du tag
ixRead	BOOL	Front montant démarre la lecture du tag
ixWrite	BOOL	Front montant démarre l'écriture sur le tag
iiRdWrStartAddr	INT	Adresse de départ dans la mémoire USER du tag pour la lecture/écriture
iiRdWrLength	INT	Nombre des octets à lire/écrire
iarbSendBuf	ANY_ARRAY_BYTE	Zone avec les données à écrire

Tableau 8

Paramètres de sortie :

Nom	Type	Description
oarbDTE102	ANY_ARRAY_BYTE	Champ de données des sorties vers le boîtier de contrôle
oxBusy	BOOL	Bloc fonctionnel occupé, aucune nouvelle commande possible
oxDone	BOOL	Fonction finie sans erreur
oxError	BOOL	Erreur détectée
owStatus	WORD	Information d'état ou code d'erreur (voir le chapitre 6.4.6)
oiDiagLength	INT	Nombre des saisies dans les données de diagnostic
oardwDiagData	ARRAY[0..3] OF DWORD	Array avec données de diagnostic (pour les codes d'erreur voir le manuel DTE102)
oxTagPresent	BOOL	Tag détecté
oiRcvdLength	INT	Nombre des octets lus
oarbRcvBuf	ANY_ARRAY_BYTE	Zone de réception pour les données lues

Tableau 9

6.4.1 Structure de la communication

Les données d'entrée du boîtier de contrôle sont mises à disposition au bloc sur le paramètre d'entrée *iarbDTE102*, le paramètre de sortie *oarbDTE102* renvoie aux données de sortie qui sont transférées au boîtier de contrôle.

Le paramètre d'entrée *iiChannelNo* détermine quelle voie du boîtier de contrôle doit être traitée.

Le paramètre d'entrée *ixConnOK* indique au bloc fonctionnel que la connexion Ethernet/IP vers le boîtier de contrôle est établie. De préférence, ce paramètre d'entrée est à lier au bit de connexion correspondant (**Figure 14**) de la zone d'entrées du module de communication.

Tant que le paramètre d'entrée *ixConnOK* est = FALSE, les paramètres de sortie *oxBusy* = TRUE et *owStatus* = 0x1000 sont fournis. Tous les autres paramètres de sortie sont initialisés avec 0.

- Mettre le paramètre d'entrée *ixConnOK* = TRUE :
La communication avec la voie paramétrée du boîtier de contrôle est activée. Les signaux pour l'échange de données sont synchronisés.
- Paramètre d'entrée *ixAntOff* = FALSE active l'antenne

Si l'échange de données a été synchronisé et l'antenne a été activée, le paramètre de sortie *oxBusy* = FALSE est fourni. Le bloc fonctionnel est maintenant prêt à échanger des données avec un tag.

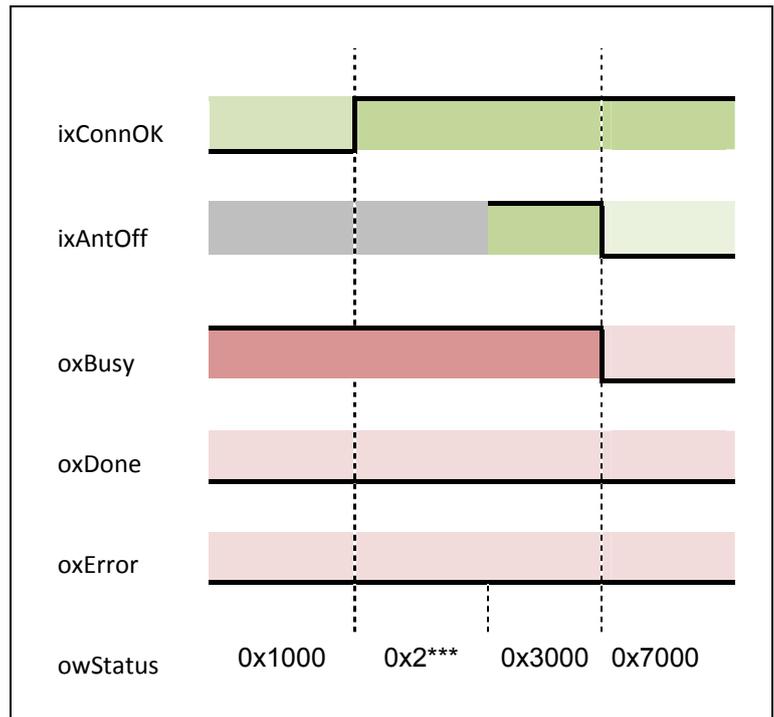


Figure 26

6.4.2 Processus du traitement de commandes

Le bloc fonctionnel DTE102_RWH_RW est un bloc fonctionnel qui travaille de manière asynchrone, c'est-à-dire que le traitement s'étend sur plusieurs appels du bloc fonctionnel.

L'état de traitement du bloc fonctionnel est indiqué sur le paramètre de sortie *owStatus*.

Tant que l'antenne raccordée détecte un tag, le paramètre de sortie *oxTagPresent* est mis.

Si le paramètre de sortie *oxBusy* est = FALSE, le bloc fonctionnel est prêt à une nouvelle commande.

- Déterminer par le paramètre d'entrée *ixUserMem* s'il y a un accès à la mémoire USER ou à l'UID du tag.
- Indiquer la zone de mémoire USER dans les paramètres d'entrée *iiRdWrStartAddr* et *iiRdWrLength*.
- Un front montant sur le paramètre d'entrée *ixRead* ou *ixWrite* démarre la lecture du tag ou l'écriture sur le tag. Les données de diagnostic indiquées sont effacées et la zone de réception définie sur le paramètre de sortie *oarbRcvBuf* est initialisée

Pendant le traitement le paramètre de sortie *oxBusy* est = TRUE. Dans le paramètre de sortie *owStatus* la progression du traitement de la commande est indiquée.

Une fois la commande terminée, le paramètre de sortie *oxBusy* = FALSE est fourni. Le résultat du traitement est communiqué dans les paramètres de sortie *oxDone* et *oxError*. Il est conservé tant que le paramètre d'entrée *ixRead* ou *ixWrite* déclenchant la commande est = TRUE, mais au moins pendant un cycle API.

Le résultat est présenté comme suit :

Aucune erreur pendant le traitement :

oxDone = TRUE

oxError = FALSE

owStatus = 0x0000

oiRcvdLength = indique le nombre des données lues

Les données lues sont rangées dans la zone de réception définie par le paramètre de sortie *oarbRcvBuf*

Erreur pendant le traitement

oxDone = FALSE

oxError = TRUE

owStatus indique un code d'erreur. Les codes d'erreur possibles sont indiqués dans le Tableau 11.

A titre d'exemple, la séquence des signaux est montrée dans la Figure 27.

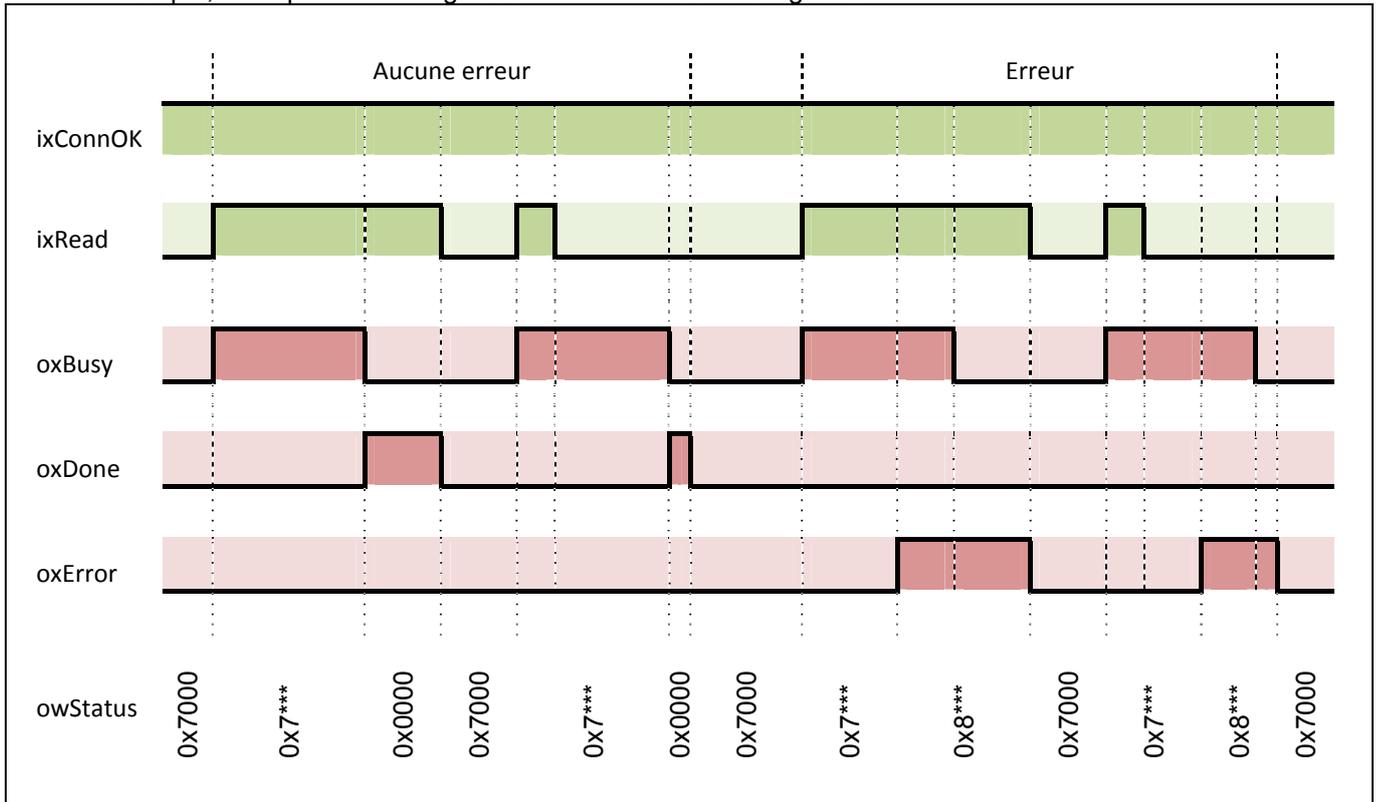


Figure 27

Si le boîtier de contrôle signale une erreur sur la voie paramétrée, le bloc fonctionnel ajoute automatiquement les signaux de commande pour la lecture des données de diagnostic dans les données de sortie vers le boîtier de contrôle. Les données de diagnostic reçues sont indiquées dans le paramètre de sortie *oardwDiagData*. Les données de diagnostic indiquées jusqu'à présent sont entièrement effacées. Le paramètre de sortie *oiDiagLength* montre combien de jeux de données de diagnostic sont indiqués actuellement.

Les codes d'erreur dans les données de diagnostic sont décrits dans le manuel du boîtier de contrôle.

6.4.3 Lecture de l'UID du tag

La manière dont le bloc fonctionnel met à disposition l'UID d'un tag dépend du paramétrage de la voie (Figure 18).

Paramètre de la voie *Edge Controlled UID Read* pas activé :

- Régler le paramètre d'entrée *ixUserMem* = FALSE

Si l'antenne détecte un tag (*oxTagPresent* = TRUE), son UID est mis à disposition dans la zone de réception. La longueur de l'UID est indiquée sur le paramètre de sortie *oiRcvdLength*.

Ces données sont effacées si le tag quitte le champ d'antenne ou si le paramètre d'entrée *ixUserMem* est = TRUE.

Paramètre de la voie *Edge Controlled UID Read* activé :

- Régler le paramètre d'entrée *ixUserMem* = FALSE
- Front montant sur le paramètre d'entrée *ixRead* démarre la lecture de l'UID du tag

Si la lecture est finie sans erreur, le paramètre de sortie *oxDone* est mis et l'UID est mis à disposition dans la zone de réception. La longueur de l'UID est indiquée sur le paramètre de sortie *oiRcvdLength*. Ces données sont conservées tant que le paramètre de sortie *oxDone* reste actif (voir le chapitre 6.4.2)

6.4.4 Lecture d'une zone de la mémoire USER du tag

Une zone de la mémoire USER d'un tag est lue comme suit :

- Régler le paramètre d'entrée *ixUserMem* = TRUE
- Indiquer l'adresse dans la mémoire USER du tag à partir de laquelle la lecture doit s'effectuer sur le paramètre d'entrée *iiRdWrStartAddr*
- Régler le nombre des octets à lire sur le paramètre d'entrée *iiRdWrLength*
- Le front montant sur le paramètre d'entrée *ixRead* applique les paramètres d'entrée *iiRdWrStartAddr* et *iiRdWrLength* et démarre la lecture depuis le tag

Les données lues depuis le tag sont rangées dans la zone de réception. Si la longueur de données indiquée ne peut pas être transférée avec un seul cycle de lecture, le bloc fonctionnel effectue automatiquement plusieurs cycles de lecture avec des offsets d'adresse correspondants dans la zone source et la zone cible. Ainsi, la longueur des données à lire est seulement limitée par le tampon de réception indiqué et la mémoire du tag.

Lorsque la commande est terminée le résultat du traitement est indiquée (voir le chapitre 6.4.2).

Dans le paramètre de sortie *oiRcvdLength* la longueur totale des données est indiquée dans la zone de réception.

Les données sont conservées tant que le paramètre de sortie *oxDone* est actif.

6.4.5 Ecriture sur une zone de la mémoire USER du tag

L'écriture sur une zone de la mémoire USER d'un tag s'effectue comme suit :

- Régler le paramètre d'entrée *ixUserMem* = TRUE
- Saisir les données à écrire dans la zone de transmission réglée sur le paramètre d'entrée *iArbSendBuf*
- Indiquer l'adresse dans la mémoire USER du tag à partir de laquelle l'écriture des données doit s'effectuer sur le paramètre d'entrée *iiRdWrStartAddr*
- Régler le nombre des octets à transférer sur le paramètre d'entrée *iiRdWrLength*
- Le front montant sur le paramètre d'entrée *ixWrite* applique les paramètres d'entrée *iiRdWrStartAddr* et *iiRdWrLength* et démarre l'écriture sur la mémoire du tag

Si la longueur de données indiquée ne peut pas être transférée avec un seul cycle d'écriture, le bloc fonctionnel effectue automatiquement plusieurs cycles d'écriture avec des offsets d'adresse correspondants dans la zone source et la zone cible. Ainsi, la longueur des données à écrire est seulement limitée par le tampon de transmission indiqué et la mémoire du tag.



Les données dans la zone de transmission ne doivent pas être changées pendant le traitement de la commande !

Lorsque la commande est terminée le résultat du traitement est indiquée (voir le chapitre 6.4.2).

6.4.6 Indications sur le paramètre de sortie *owStatus*

Indications d'état

owStatus	Description
0x0000	Fonction terminée sans erreur
0x1000	Attendre le paramètre d'entrée <i>ixConnOK</i> = TRUE
0x2000	Pendant le démarrage vérifier des signaux pour l'échange de données
0x2101	Pendant le démarrage activer le mode de la mémoire USER
0x2102	Pendant le démarrage synchroniser la lecture de la mémoire USER
0x2103	Pendant le démarrage finir la lecture de la mémoire USER
0x2201	Pendant le démarrage activer le mode de la mémoire USER
0x2202	Pendant le démarrage synchroniser l'écriture sur la mémoire USER
0x2203	Pendant le démarrage finir l'écriture sur la mémoire USER
0x2402	Pendant le démarrage synchroniser la lecture des données de diagnostic
0x2403	Pendant le démarrage finir la lecture des données de diagnostic
0x3000	Attendre le paramètre d'entrée <i>ixAntOff</i> = FALSE
0x7000	Attendre le démarrage d'une commande
0x7101	Activer le mode de la mémoire USER
0x7102	Lecture de la mémoire USER
0x7103	Finir la lecture de la mémoire USER
0x7201	Activer le mode de la mémoire USER
0x7202	Ecriture sur la mémoire USER
0x7203	Finir l'écriture sur la mémoire USER
0x7301	Activer l'accès à l'UID
0x7302	Lecture de l'UID
0x7303	Finir la lecture de l'UID
0x7401	Finir le passage à la mémoire USER ou à l'UID
0x7402	Lecture des données de diagnostic
0x7403	Finir la lecture des données de diagnostic
0x8***	Code d'erreur (voir le Tableau 11)

Tableau 10

Codes d'erreur

owStatus	Description
0x8000	Le boîtier de contrôle signale une erreur ; évaluer les données de diagnostic dans le paramètre de sortie <i>oardwDiagData</i>
0x8101 0x8102 0x8103 0x8104 0x8105 0x8106	Paramètre d'entrée <i>iiChannelNo</i> non valable Paramètre d'entrée <i>iardDTE102</i> non valable Paramètre d'entrée <i>oarbDTE102</i> non valable Différentes longueurs du paramètre d'entrée <i>iardDTE102</i> et du paramètre de sortie <i>oarbDTE102</i> Paramètre d'entrée <i>iardSendBuf</i> non valable Paramètre de sortie <i>oarbRcvBuf</i> non valable
0x8201 0x8202	Paramètre d'entrée <i>ixConnOK</i> = FALSE pendant la transmission des données Timeout ; le boîtier de contrôle ne réagit pas
0x8301 0x8302 0x8303 0x8304 0x8305 0x8306 0x8307	Démarrage d'une commande pendant que le bloc fonctionnel est occupé (<i>oxBusy</i> = TRUE) Démarrage d'une commande avec une antenne désactivée (<i>ixAntOff</i> = TRUE) Démarrage simultané des commandes de lecture et d'écriture Démarrage d'une commande sans tag dans le champ d'antenne Paramètre d'entrée <i>iiRdRwLength</i> <= 0 ou supérieur à la zone de réception définie (<i>oarbRcvBuf</i>) Paramètre d'entrée <i>iiRdRwLength</i> <= 0 ou supérieur à la zone de transmission définie (<i>iardSendBuf</i>) L'écriture de l'UID n'est pas supportée par le système
0x8401	Zone de réception (<i>oarbRcvBuf</i>) trop petite pour la lecture cyclique de l'UID

Tableau 11