

Interface de commande intuitive  
Automate Modicon M340

Description de la bibliothèque

**DTE810**

**DTE910**

## Contenu

1	Synoptique.....	3
2	Exclusion de la responsabilité .....	3
3	Réglage des paramètres de connexion dans le lecteur RFID .....	3
4	Projeter la connexion dans Unity Pro .....	5
4.1	Ajouter le lecteur RFID dans le projet logiciel.....	6
4.2	Réglage des entrées et sorties TOR du lecteur RFID comme connexion cible .....	7
4.3	Réglage des paramètres de connexion dans le module de communication.....	8
5	Travailler avec la bibliothèque.....	9
5.1	Intégrer la bibliothèque dans Unity Pro .....	9
5.2	Contenu de la bibliothèque .....	10
5.3	Utiliser les blocs fonctionnels dans le programme d'application .....	11
6	Le bloc fonctionnel DTE810_910expl en détail.....	11
6.1	Description de la fonction .....	11
6.2	Fonctionnement du bloc fonctionnel DTE810_910expl.....	14
6.2.1	Etablir la connexion au lecteur RFID .....	14
6.2.2	Exécuter une fonction .....	15
6.3	Descriptions des fonctions.....	17
6.3.1	ID fonction 1 - Lecture de l'EPC de tous les tags.....	18
6.3.2	ID fonction 2 - Lecture de la mémoire TID de tous les tags .....	19
6.3.3	ID fonction 3 - Lecture de la mémoire TID d'un tag défini .....	20
6.3.4	ID fonction 4 - Lecture d'une zone de la mémoire USER de tous les tags .....	21
6.3.5	ID fonction 5 - Lecture d'une zone de la mémoire USER d'un tag défini .....	23
6.3.6	ID fonction 6 - Ecriture sur une zone de la mémoire USER de tous les tags.....	24
6.3.7	ID fonction 7 - Ecriture sur une zone de la mémoire USER d'un tag .....	26
6.3.8	ID fonction 8 – Ecriture d'un EPC sur un tag inconnu .....	27
6.3.9	ID fonction 9 - Modifier l'EPC d'un tag.....	29
6.3.10	ID fonction 15 - Réglage de la puissance d'antenne.....	30
6.4	Déconnexion et reconnexion de la communication avec le lecteur RFID .....	31
6.5	Indications d'état sur l'entrée owStatus .....	32
7	Codes d'erreur.....	33
7.1	Codes d'erreur sur l'entrée owStatus .....	33
7.2	Codes d'erreur des données de tag .....	34
7.3	Codes d'erreur d'accès au tag.....	34
8	Description des types de données dérivés .....	34
8.1	Arrays.....	34
8.2	Structures.....	35

## 1 Synoptique

La bibliothèque DTE810 / DTE910 pour Unity Pro contient un package de codes pour l'automate Modicon M340 de Schneider Electric pour simplifier l'échange de données avec le lecteur RFID DTE810 ou DTE910 d'ifm electronic dans le programme d'application de l'API.

Les fonctions suivantes sont fournies par les blocs fonctionnels de la bibliothèque dans le programme d'applications:

Lecture de l'EPC de tous les tags dans le champ d'antenne

Ecriture d'un EPC sur un tag

Lecture de la mémoire TID de tous les tags dans le champ d'antenne ou d'un tag sélectionné

Lecture d'une zone de la mémoire USER de tous les tags dans le champ d'antenne ou d'un tag sélectionné

Ecriture sur une zone de la mémoire USER de tous les tags dans le champ d'antenne ou d'un tag sélectionné

Réglage de la puissance d'antenne

Ce document détaille la configuration de la connexion entre l'automate et le lecteur RFID 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 Réglage des paramètres de connexion dans le lecteur RFID

Pour régler les paramètres de connexion dans le lecteur RFID, le logiciel ReaderStart version 2.30 ou supérieur est nécessaire.

- Démarrer le logiciel ReaderStart sur le PC et établir une connexion avec le lecteur RFID.

- Cliquer sur le bouton [v] *Module intelligence configuration de la communication* pour ouvrir les paramètres du port.

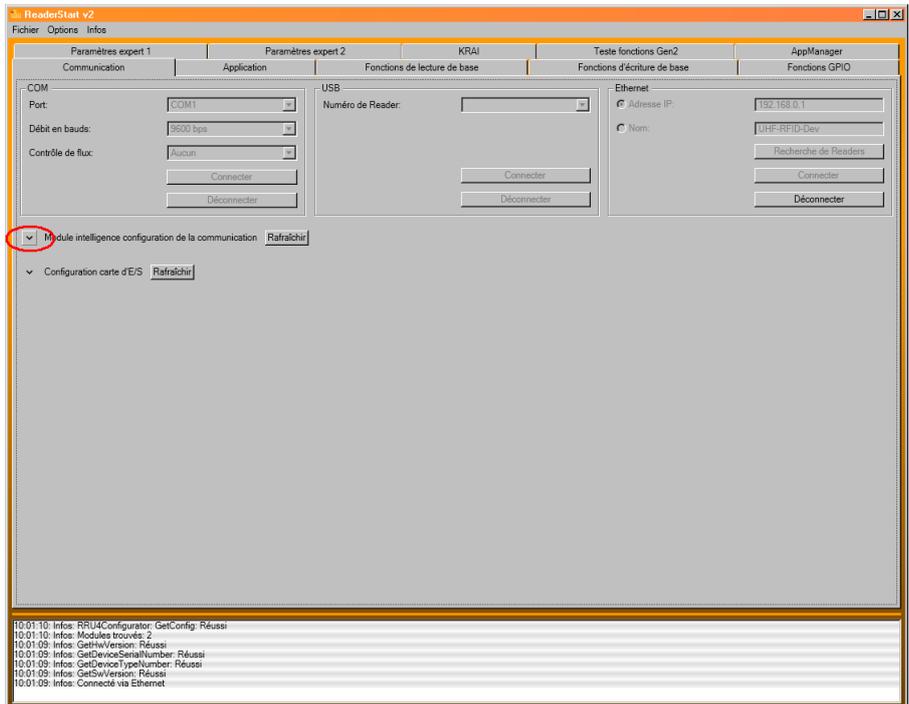


Figure 1

- Dans le secteur *Paramètres de port*, régler les paramètres du port de communication.
- Pour les appliquer dans le lecteur RFID, cliquer sur [Enregistrer les modifications].

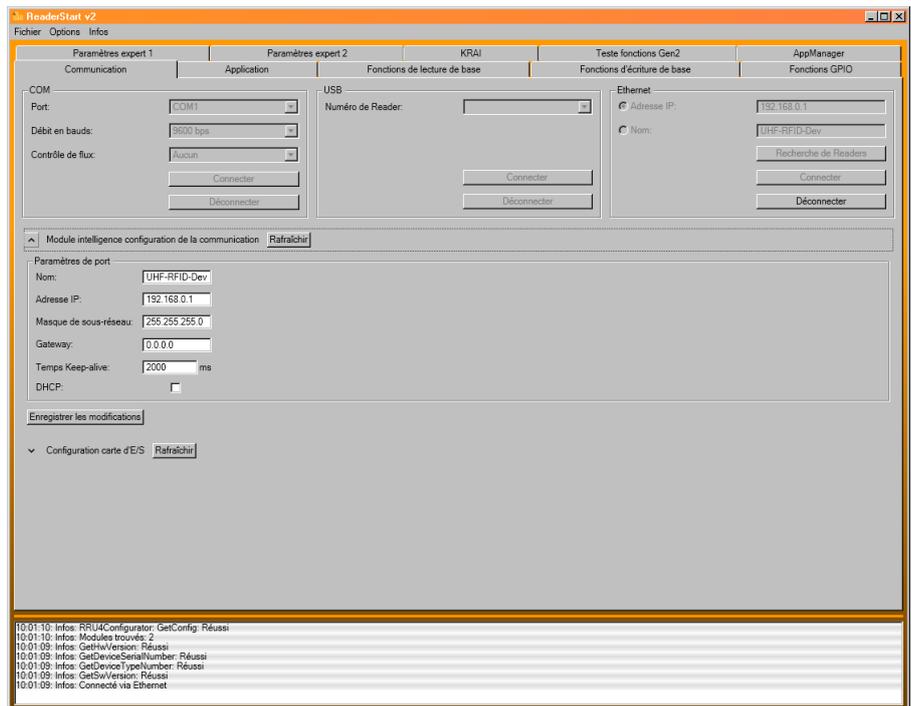


Figure 2

#### 4 Projeter la connexion dans Unity Pro

La communication entre l'automate et le lecteur RFID s'effectue via Ethernet/IP grâce à un module de communication NOC0401 dans l'automate M340. Le bloc de fonction de la bibliothèque DTE810 / DTE910 utilise le service acyclique "Explicit Messaging" pour l'échange de données avec le lecteur RFID. Aucune projection n'est nécessaire pour une connexion.

Cependant il est recommandé de projeter une connexion cyclique au lecteur RFID. Ainsi il est possible de déterminer facilement l'état de la connexion Ethernet/IP au lecteur RFID en évaluant le bit Health de la connexion fourni par le module de communication.

Mais une connexion cyclique occupe les assemblages affectés de sorte qu'un accès acyclique n'est plus possible. C'est pourquoi il est recommandé de projeter une connexion cyclique dans l'automate aux entrées et sorties TOR du lecteur RFID. Ainsi tous les assemblages des ports d'antenne sont disponibles pour le service acyclique.

Les entrées et sorties TOR du lecteur RFID utilisent 2 octets dans la zone d'entrées et de sorties du module de communication NOC0401. Il faut en prendre compte pour la configuration de l'adressage du module de communication.

#### 4.1 Ajouter le lecteur RFID dans le projet logiciel

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



Figure 3

- Sélectionner *Generic Device* de *Schneider Electric* de la liste des équipements.
- Cliquer sur [Ajouter DTM] pour ouvrir la fenêtre *Propriétés de l'équipement*.

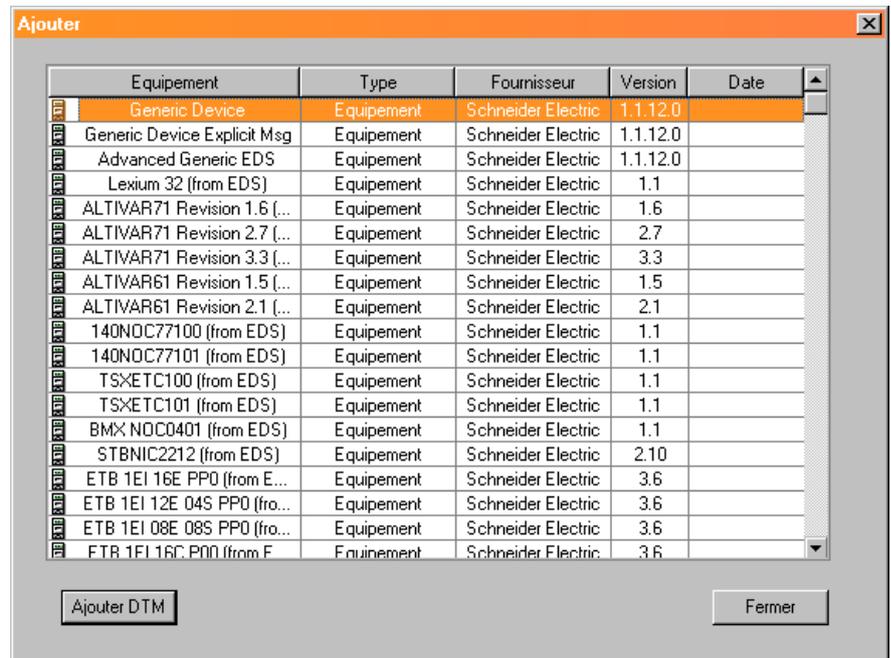


Figure 4

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

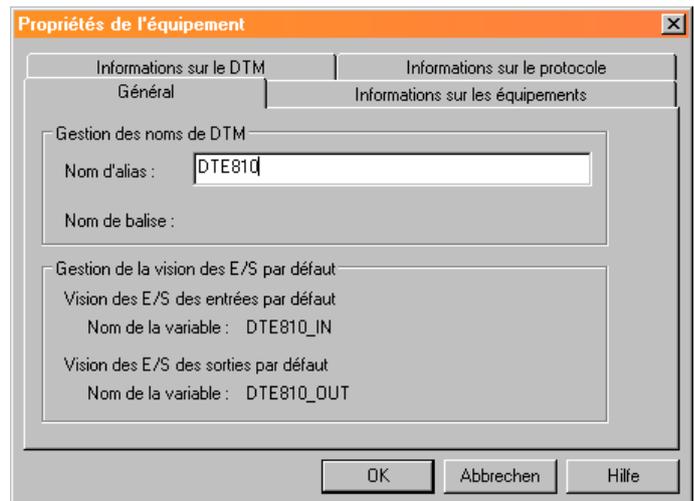


Figure 5

- 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.

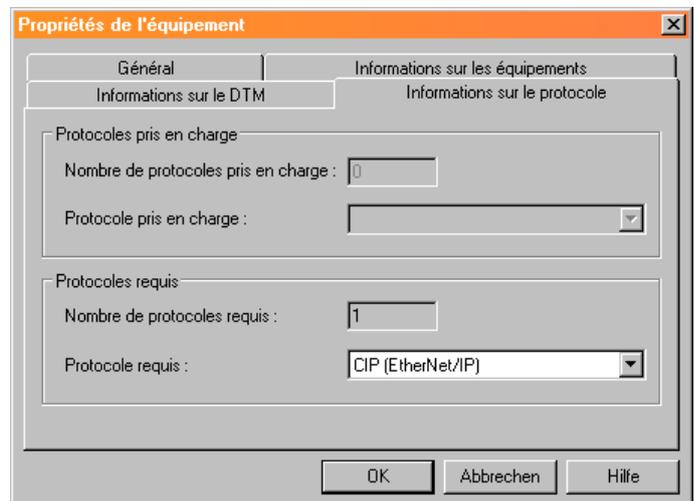


Figure 6

#### 4.2 Réglage des entrées et sorties TOR du lecteur RFID comme connexion cible

- Dans le navigateur de DTM cliquer sur l'équipement ajouté par le bouton droit de la souris.
- Sélectionner via le menu contextuel *Ouvrir* pour ouvrir la fenêtre *fdtConfiguration*.

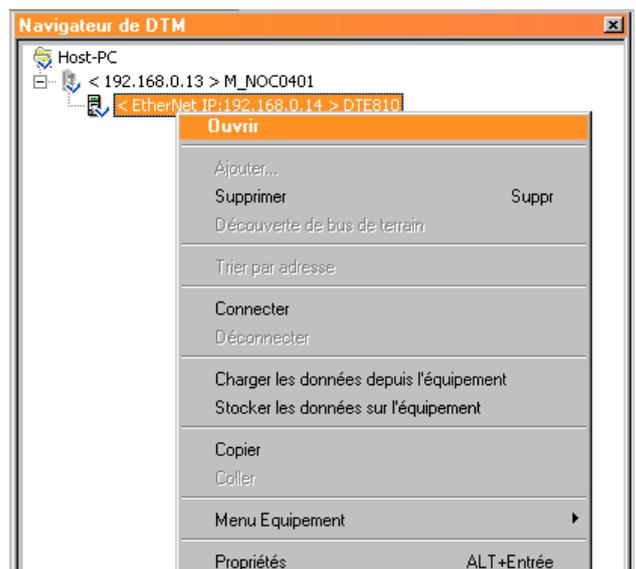


Figure 7

- Dans la structure cliquer sur *Exclusive Owner*.
- Dans l'onglet *Général* régler les assemblages de communication pour l'échange de données cycliques.
- Régler les valeurs suivantes :  
 Taille des entrées = 2  
 Input Instance = 101  
 Tailles des sorties = 2  
 Output Instance = 100
- Dans les onglets *Validation d'identité* et *Paramètres de configuration* aucune modification n'est nécessaire.
- Cliquer sur [OK] pour appliquer les données.

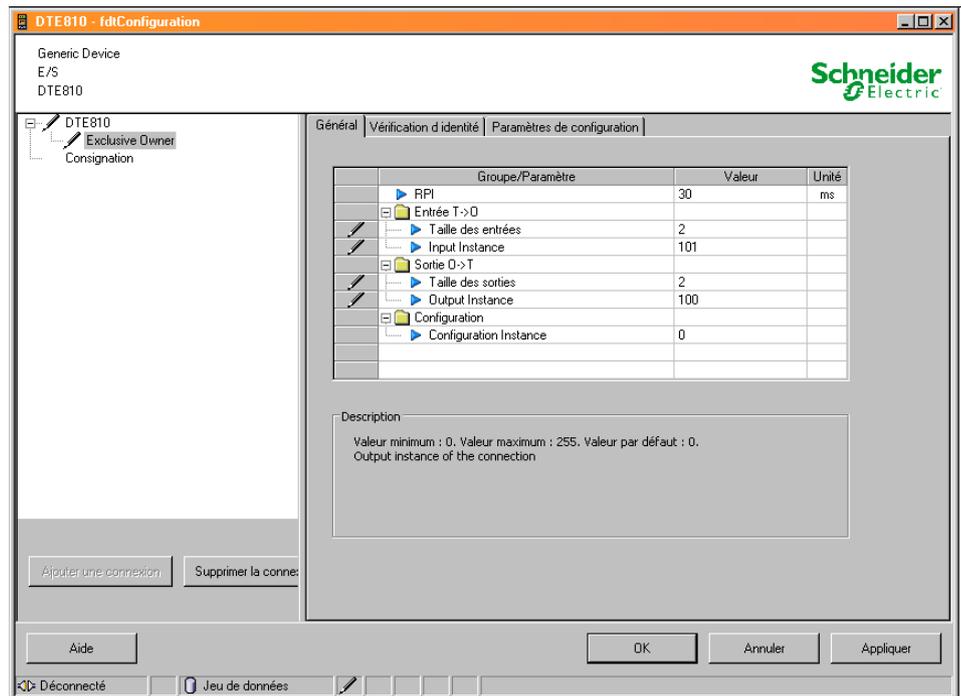


Figure 8

#### 4.3 Réglage des paramètres de connexion dans le module de communication

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



Figure 9

- Dans la structure étendre la saisie *Liste des équipements* et sélectionner le lecteur RFID.
- Dans l'onglet *Paramétrage de l'adresse* saisir les valeurs réglées dans le lecteur RFID.
- Cliquer sur [OK] pour appliquer les réglages.

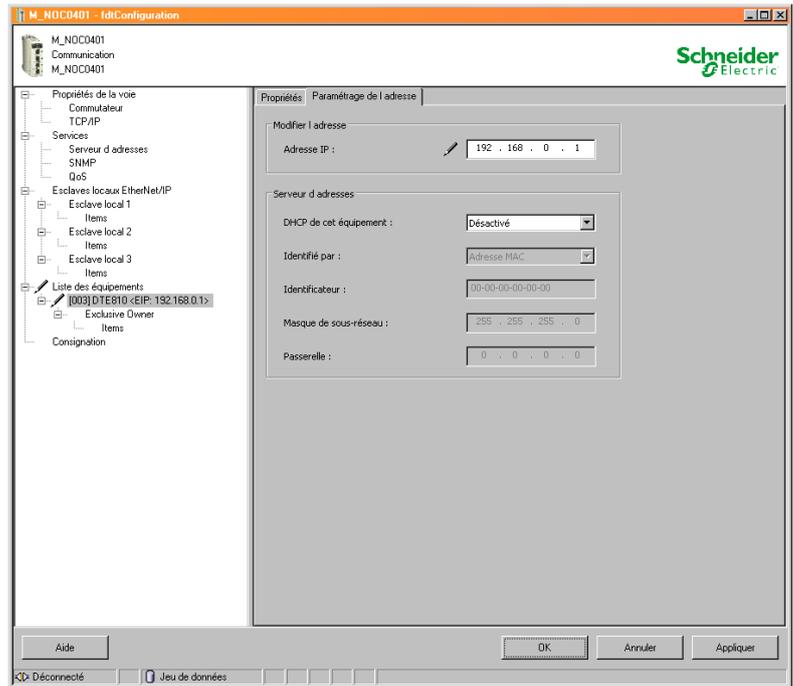


Figure 10

## 5 Travailler avec la bibliothèque

### 5.1 Intégrer la bibliothèque dans Unity Pro

La bibliothèque DTE810 /DTE910 est fournie comme fichier d'archive. Après le décompactage de cette archive la bibliothèque doit 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.

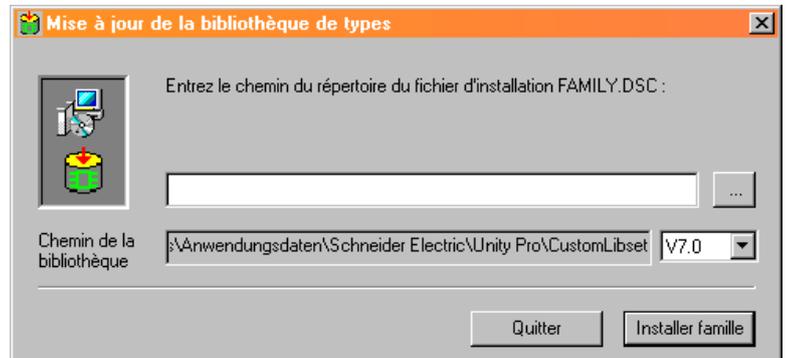


Figure 11

Dans la bibliothèque ifm electronic les blocs sont rangés dans la famille DTE810 / DTE910.

Library	Description
<Libset V7.0>	
Base Lib	This is the main manufacturer library. It contains the main IEC families and some o
Communication	This library is used for exchanging data between devices connected by a bus or r
CONT_CTL	This library is for projecting process-engineering servo-loops. It contains Condition
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 contain
I/O Management	This library contains EFBs, which are required for using I/O modules. It contains t
ifm electronic	This library contains DFBs and DDTs for devices made by ifm electronic.
DTE810 / DTE910	This library contains DFBs and DDTs for the RFID reader DTE810 and DTE910.
MotionFunctionBlock	This library contains Motion Function Blocks.
Motion	This Library contains the Axis Control, CAM control, Lexium and MMF Start Familie
Obsolete Lib	This library is used for compatibility with PL7 or Concept applications. It contains C
Safety	This library contains safety EF/EFBs for the usage in safety applications.
System	This library contains Events, SFC Management, SysClock Families.
Unitid L984	L984 Library for Unity Pro. Contains EFBs for exclusive usane in L984 editor or

Figure 12

## 5.2 Contenu de la bibliothèque

La bibliothèque DTE810 / DTE910 est composée d'un bloc de fonction central DTE810\_910expl et d'un groupe de types de données spéciaux dont le bloc fonctionnel a besoin (Tableau 1).

Nom	Type	Description
DTE810_910expl	DFB	Bloc code central de la bibliothèque
Dte810arstTags	ARRAY[1..n] OF Dte810stTagData	Array pour la réception des données des tags traités L'array doit commencer par l'index 1. L'index final n définit combien de tags max peuvent être traités par le DFB. Il doit être configuré pour être compatible avec l'application.
Dte810arwTagEpc	ARRAY[0..n] OF WORD	Array pour la réception de l'EPC d'un tag L'array doit commencer par l'index 0. L'index final n définit la longueur maximale d'un EPC que le DFB peut traiter. Il doit être configuré pour être compatible avec l'application.
Dte810arwTagMemory	ARRAY[0..n] OF WORD	Array pour la réception des données mémoire TID ou USER d'un tag L'array doit commencer par l'index 0. L'index final n définit la taille mémoire maximale que le DFB peut traiter. Il doit être configuré pour être compatible avec l'application.
Dte810stAntPortPower	Struct	Données pour régler la puissance d'antenne
Dte810stDiagData	Struct	Informations détaillées sur l'erreur
Dte810stDteSet	Struct	Réglages de la connexion au lecteur RFID
Dte810stNocSet	Struct	Localisation du module de communication NOC0401 dans l'automate
Dte810stReadEpc	Struct	Paramètre pour la lecture de tous les EPC
Dte810stReadUserData	Struct	Paramètre pour la lecture de la mémoire USER
Dte810stSettings	Struct	Données pour établir une connexion au lecteur RFID
Dte810stTagData	Struct	Données lues par un tag
Dte810stTagEpcData	Struct	Données d'un EPC
Dte810stTagMemoryData	Struct	Données TID ou USER
Dte810stWriteUserData	Struct	Données pour l'écriture de la mémoire USER d'un tag

Tableau 1

Les structures sont décrites dans le chapitre 8.2. Les significations des éléments individuels des structures et leur utilisation sont expliqués dans les chapitres 6.6.2 et 6.3

### 5.3 Utiliser les blocs fonctionnels dans le programme d'application

Si une instance du bloc fonctionnel DTE810\_910expl est créée à partir de la bibliothèque dans le programme d'application, les types de données utilisés sont aussi créés automatiquement dans le programme d'application.

Seulement une instance du bloc fonctionnel par numéro de port d'antenne d'un lecteur RFID est permissible. Toutes les instances utilisées doivent être appelées une fois par cycle dans le programme cyclique de l'API.

La taille de la mémoire nécessaire d'une instance dans le programme d'utilisation est influencée par les tailles des arrays *Dte810arstTags*, *Dte810arwTagEpc* et *Dte810arwTagMemory*. Elles peuvent être adaptées aux exigences de l'application. Seulement l'index final de l'array peut être modifié, l'index de début et le type de données doivent restés sur leurs valeurs initiales.

Un exemple pour illustration :

- Un maximum de 7 tags simultanément est à attendre.  
→ Déclaration *Dte810arstTags* : ARRAY[1..7] OF Dte810stTagData
- L'EPC le plus long est de 6 mots.  
→ Déclaration *Dte810arwTagEpc* : ARRAY[0..5] OF WORD
- Un maximum de 12 mots de données est à lire ou écrire.  
→ Déclaration *Dte810arwTagMemory* : ARRAY[0..11] OF WORD

## 6 Le bloc fonctionnel DTE810\_910expl en détail

### 6.1 Description de la fonction

La communication doit s'effectuer selon une procédure définie. Ceci est montré à l'aide d'un diagramme dans la **Figure Figure 13**.

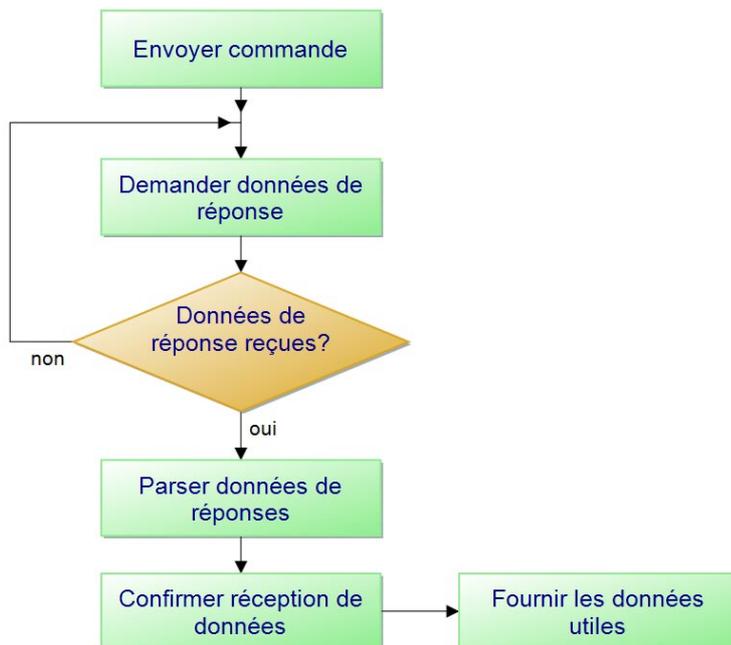


Figure 13

Pour rendre le traitement de la commande plus convivial, le bloc fonctionnel *DTE810\_910expl* a été créé. Ce bloc traite automatiquement le protocole de commande montré ci-dessus.

L'image suivante montre le bloc fonctionnel DTE810\_910expl :

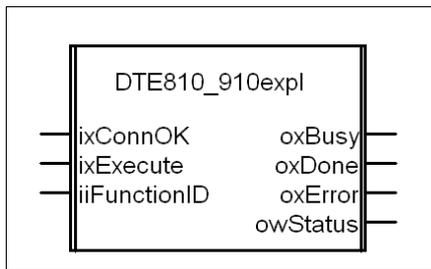


Figure 14

Description des paramètres :

Entrées :

Nom	Type	Description
ixConnOK	BOOL	Connexion antenne - lecteur RFID - API établie
ixExecute	BOOL	Front montant active la fonction réglée sur iiFunctionID
iiFunctionID	INT	ID de la fonction qui doit être exécutée (cf. Tableau 6)

Tableau 2

Sorties:

Nom	Type	Description
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 (cf. Tableau 27)

Tableau 3

Variables publiques :

Nom	Type	Description
stSettings	Dte810stSettings	Données pour le réglage de base de la connexion au lecteur RFID
stReadEpc	Dte810stReadEpc	Paramètre pour l'écriture des EPC
stReadUserData	Dte810stReadUserData	Paramètre pour la lecture de la mémoire USER du tag
stSelectionEpc	Dte810stTagEpcData	Données EPC pour la sélection d'un tag défini
dwTagPassword	DWORD	Mot de passe pour l'accès au tag
stEpcToWrite	Dte810stTagEpcData	Paramètre pour l'écriture d'un nouveau EPC sur un tag
stUserDataToWrite	Dte810stWriteUserData	Paramètre et données pour l'écriture dans la mémoire USER des tags
stAntPortPower	Dte810stAntPortPower	Paramètre pour le réglage de la puissance d'antenne
stDiagData	Dte810stDiagData	Données de diagnostic
iTagCounter	INT	Nombre des tags trouvés
arTag	Dte810arstTags	Données lues des tags

Tableau 4

## 6.2 Fonctionnement du bloc fonctionnel DTE810\_910expl

### 6.2.1 Etablir la connexion au lecteur RFID

Avant que la communication avec le lecteur RFID puisse démarrer, tous les réglages et paramètres de base dans la structure *stSettings* doivent être saisis dans les variables publiques de l'instance (Tableau 5).

Nom	Type de données	Valeur de saisie
stSettings	Dte810stSettings	Données pour établir une connexion au lecteur RFID
stNOC	Dte810stNocSet	Localisation du module de communication NOC0401 dans L'API
iRackNo	INT	Numéro du rack dans lequel le module de communication est monté
iSlotNo	INT	Numéro du slot dans lequel le module de communication est monté
iChannelNo	INT	Numéro de la voie du module de communication
stDTE	Dte810stDteSet	Réglages de la connexion au lecteur RFID
arilPAddr	ARRAY[1..4] OF INT	Adresse IP du lecteur RFID Exemple : Adresse IP = 192.168.0.1  arilPAddr[1] := 192 arilPAddr[2] := 168 arilPAddr[3] := 0 arilPAddr[4] := 1
iAntPortNo	INT	= 0 échange de données avec tous les ports d'antenne activés 1 .. 4 échange de données avec le port d'antenne 1 .. 4
iTimeOutVal	INT	= 0 fonction interne DATA_EXCH utilise le temps de contrôle standard > 0 temps de contrôle de la fonction interne DATA_EXCH (valeur * 100ms)
iRepeatTime	INT	Temporisation pour répéter la commande quand le lecteur RFID ne fournit pas de données (recommandation = 200 ms)

Tableau 5

Le réglage de l'entrée *ixConnOK* = TRUE active la communication avec le lecteur RFID.

Les valeurs de la structure *stSettings* dans les variables publiques sont adoptées.

Dans les variables publiques les données dans l'array *arTag* et le compteur tag *iTagCounter* prennent la valeur 0.

Le bloc fonctionnel demande des données du lecteur RFID cycliquement. Quand des données peuvent être reçues, le bloc fonctionnel synchronise l'échange de données.

A la fin du traitement la sortie *oxBusy* = FALSE est envoyé. Le bloc fonctionnel est prêt à exécuter une fonction. La séquence des signaux est montrée dans la Figure 15.

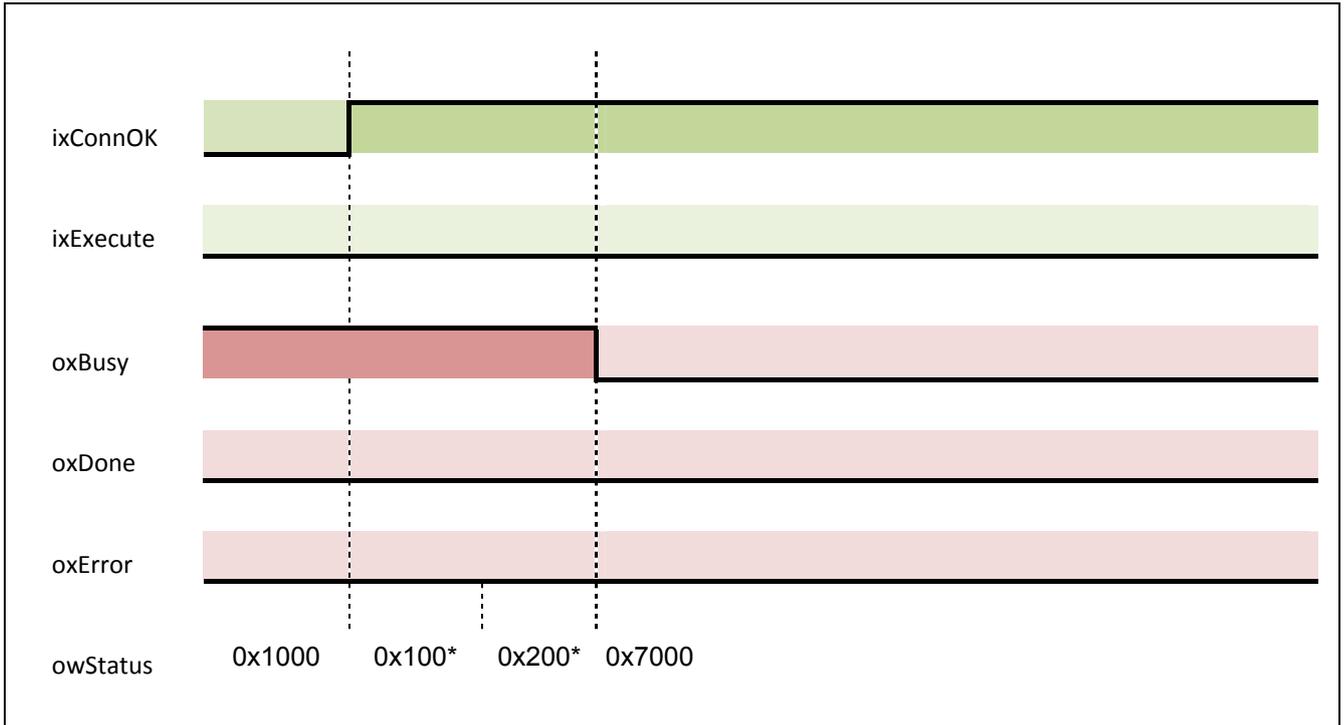


Figure 15

### 6.2.2 Exécuter une fonction

Le bloc fonctionnel DTE810\_910expl est un bloc fonctionnel qui travaille de manière asynchrone, c'est-à-dire que le traitement s'étend sur plusieurs appels du bloc fonctionnel.

Une fonction à exécuter est sélectionnée en indiquant l'ID de fonction sur le paramètre *iiFunctionID*. Les paramètres et données de la fonction nécessaires doivent être saisis dans la structure respective dans les données publiques du bloc fonctionnel. La fonction est démarrée avec le front montant sur l'entrée *iiExecute*.

L'état de la commande est affiché via les paramètres de sortie *oxBusy* et *owStatus*.

Pendant le traitement de la commande le paramètre *oxBusy* est activé. Le paramètre *owStatus* indique la progression du traitement de la commande. A la fin de la commande le résultat est fourni dans les paramètres *oxDone* et *oxError* et y reste jusqu'à ce que l'entrée *ixExecute* = TRUE, au moins pendant un cycle API.

Si aucune erreur ne s'est produite, *oxDone* est activé et la valeur 0x0000 est envoyé dans le paramètre *owStatus*. Lorsqu'il y a des données de réponse du tag aux commandes, ces données sont fournies dans l'array *arTag* dans les variables publiques. La variable publique *iTagCounter* indique combien de tags ont été détectés ou influencés par la fonction.

Si une erreur se produit pendant le traitement de la commande, *oxError* est activé. Pour une description plus détaillée de l'erreur qui s'est produite, un code d'erreur est indiqué dans le paramètre *owStatus*. Les codes d'erreur possibles sont indiqués dans le Tableau 28.

La séquence des signaux est indiquée dans la Figure 16.

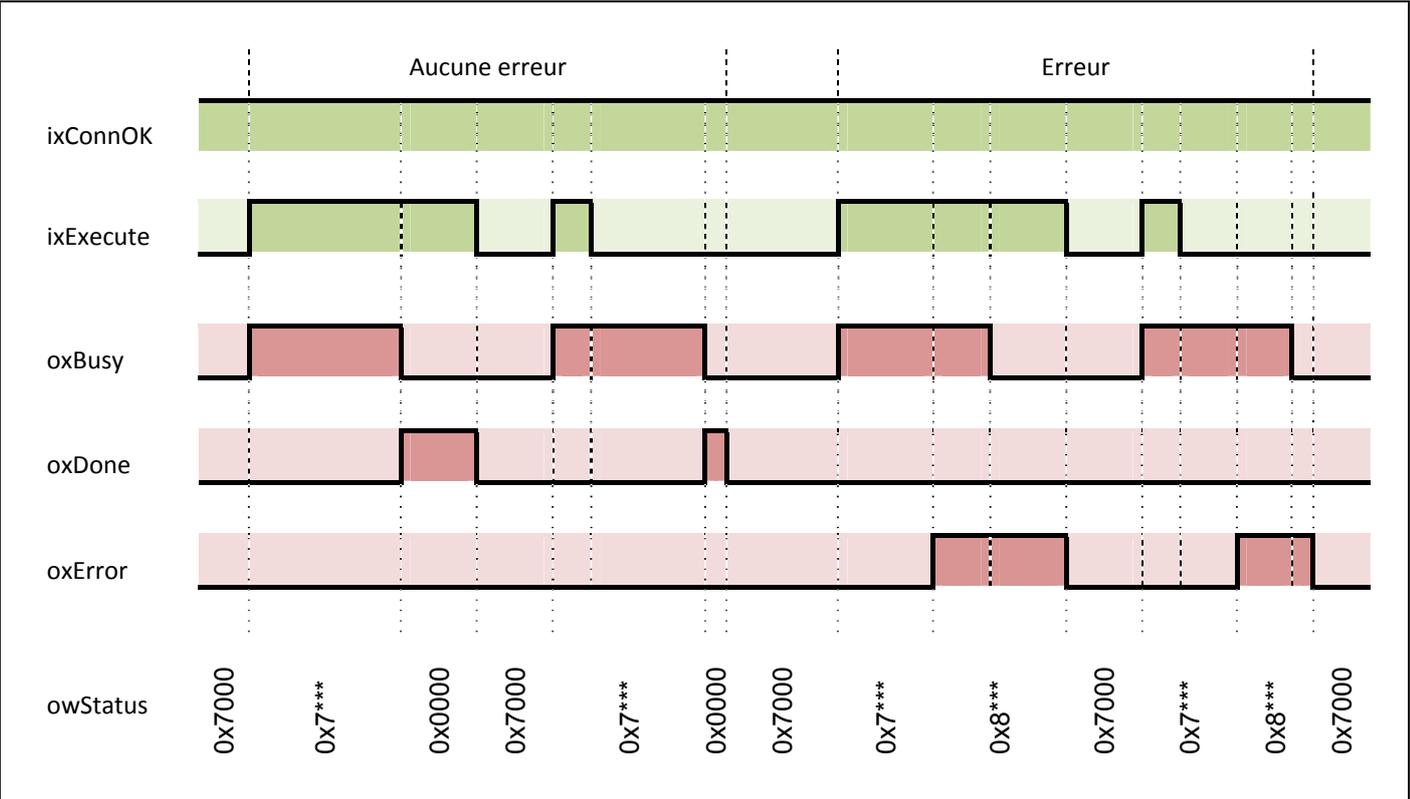


Figure 16

### 6.3 Descriptions des fonctions

Fonction	ID (décimale)	Paramètres nécessaires	Données de réponse	Description
Lecture de L'EPC de tous les tags	1	stReadEpc	iTagCounter, arTag	Cf. chapitre 6.3.1
Lecture de la mémoire TID de tous les tags	2	dwTagPassword	iTagCounter, arTag	Cf. chapitre 6.3.2
Lecture de la mémoire TID d'un tag	3	dwTagPassword, stSelectionEpc	iTagCounter, arTag	Cf. chapitre 6.3.3
Lecture d'une zone de la mémoire USER de tous les tags	4	dwTagPassword, stReadUserData	iTagCounter, arTag	Cf. chapitre 6.3.4
Lecture d'une zone de la mémoire USER depuis un tag	5	dwTagPassword, stSelectionEpc, stReadUserData	iTagCounter, arTag	Cf. chapitre 6.3.5
Ecriture sur une zone de la mémoire USER de tous les tags	6	dwTagPassword, stUserDataToWrite	iTagCounter, arTag	Cf. chapitre 6.3.6
Ecriture sur une zone de la mémoire USER d'un tag	7	dwTagPassword, stSelectionEpc, stUserDataToWrite	iTagCounter, arTag	Cf. chapitre 6.3.7
Ecriture de l'EPC sur un tag inconnu	8	dwTagPassword, stEpcToWrite	iTagCounter, arTag	Cf. chapitre 6.3.8
Modifier l'EPC d'un tag	9	dwTagPassword, stSelectionEpc, stEpcToWrite	iTagCounter, arTag	Cf. chapitre 6.3.9
Réglage de la puissance d'antenne	15	stAntPortPower		Cf. chapitre 6.3.10

Tableau 6

### 6.3.1 ID fonction 1 - Lecture de l'EPC de tous les tags

Cette fonction crée une liste avec les EPC de tous les tags qui ont été détectés pendant le temps de lecture réglé, indépendamment si les tags se trouvent dans le champ d'antenne ou non après la finrenvoi de la lecture.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
stReadEpc	Dte810stReadEpc	Paramètre pour la lecture de tous les EPC
iReadingTime	INT	= 0 lecture unique des EPC de tous les tags > 0 temps d'activation (en ms) pour la lecture des EPC de tous les tag

Tableau 7

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	Nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	0x00
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	Données lues du tag ...
arTag[n]	DTE810stTagData	Données lues du tag n

Tableau 8

### 6.3.2 ID fonction 2 - Lecture de la mémoire TID de tous les tags

Cette fonction lit la mémoire TID de tous les tags dans le champ d'antenne dont le mot de passe pour l'accès est identique au mot de passe saisi. Pour les tags, le mot de passe 0x0000 0000 est réglé par défaut.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (mémoire sur le tag)

Tableau 9

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	Nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	Données TID lues
iLength	INT	Nombre des mots de données TID
arwDataWord[0]	WORD	Mot de données TID 0 lu
arwDataWord[...]	WORD	Mot de données TID ... lu
arwDataWord[n]	WORD	Mot de données TID n lu
arTag[...]	DTE810stTagData	Données lues du tag ...
arTag[n]	DTE810stTagData	Données lues du tag n

Tableau 10

### 6.3.3 ID fonction 3 - Lecture de la mémoire TID d'un tag défini

Cette fonction lit la mémoire TID d'un tag défini. Le tag à lire est sélectionné via l'EPC. Le mot de passe du tag est aussi à saisir. Pour le tag, le mot de passe 0x0000 0000 est réglé par défaut.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (méorisé sur le tag)
stSelectionEPC	DTE810stTagEPCData	Données EPC pour la sélection d'un tag
iLength	INT	Nombre des mots de données EPC du tag sélectionné
arwEpcWord[0]	WORD	Mot de données EPC 0 du tag sélectionné
arwEpcWord[...]	WORD	Mot de données EPC ... du tag sélectionné
arwEpcWord[n]	WORD	Mot de données EPC n du tag sélectionné

Tableau 11

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	0 .. nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	Données TID lues
iLength	INT	Nombre des mots de données TID
arwDataWord[0]	WORD	Mot de données TID 0 lu
arwDataWord[...]	WORD	Mot de données TID ... lu
arwDataWord[n]	WORD	Mot de données TID n lu
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tableau 12

#### 6.3.4 ID fonction 4 - Lecture d'une zone de la mémoire USER de tous les tags

Cette fonction lit la zone de la mémoire USER de tous les tags dans le champ d'antenne à indiquer dont le mot de passe accès est identique au mot de passe saisi. Pour les tags, le mot de passe 0x0000 0000 est réglé par défaut.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (mémoire sur le tag)
stReadUserData	DTE810stReadUserData	Paramètre pour la lecture de la mémoire USER
diStartAddress	DINT	Adresse mémoire à partir de laquelle la lecture doit s'effectuer
iLength	INT	Nombre des mots de données à lire

Tableau 13

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	Nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	Données mémoire USER lues
iLength	INT	Nombre des mots de données mémoire USER
arwDataWord[0]	WORD	Mot de données mémoire USER 0 lu
arwDataWord[...]	WORD	Mot de données mémoire USER ... lu
arwDataWord[n]	WORD	Mot de données mémoire USER n lu
arTag[...]	DTE810stTagData	Données lues du tag ...
arTag[n]	DTE810stTagData	Données lues du tag n

Tableau 14

### 6.3.5 ID fonction 5 - Lecture d'une zone de la mémoire USER d'un tag défini

Cette fonction lit une zone à indiquer de la mémoire USER d'un tag défini. Le tag à lire est sélectionné via l'EPC. Le mot de passe du tag est aussi à saisir. Pour le tag, le mot de passe 0x0000 0000 est réglé par défaut.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (mémorisé sur le tag)
stSelectionEPC	DTE810stTagEPCData	Données EPC pour la sélection d'un tag
iLength	INT	Nombre des mots de données EPC du tag sélectionné
arwEpcWord[0]	WORD	Mot de données EPC 0 du tag sélectionné
arwEpcWord[...]	WORD	Mot de données EPC ... du tag sélectionné
arwEpcWord[n]	WORD	Mot de données EPC n du tag sélectionné
stReadUserData	DTE810stReadUserData	Paramètre pour la lecture de la mémoire USER
diStartAddress	DINT	Adresse mémoire à partir de laquelle la lecture doit s'effectuer
iLength	INT	Nombre des mots de données à lire

Tableau 15

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	0 .. nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	Données mémoire USER lues
iLength	INT	Nombre des mots de données mémoire USER
arwDataWord[0]	WORD	Mot de données mémoire USER 0 lu
arwDataWord[...]	WORD	Mot de données mémoire USER ... lu
arwDataWord[n]	WORD	Mot de données mémoire USER n lu
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tableau 16

### 6.3.6 ID fonction 6 - Ecriture sur une zone de la mémoire USER de tous les tags

Cette fonction écrit des données dans une zone à indiquer de la mémoire USER de tous les tags dans le champ d'antenne dont le mot de passe accès est identique au mot de passe saisi. Pour les tags, le mot de passe 0x0000 0000 est réglé par défaut.

S'il y a trop de tags dans le champ d'antenne, aucune donnée n'est écrite et une erreur est retournée.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (mémoire sur le tag)
stUserDataToWrite	Dte810WriteUserData	Paramètre pour écrire des données de la mémoire USER
diStartAddress	DINT	Adresse mémoire à partir de laquelle l'écriture doit s'effectuer
iLength	INT	Nombre des mots de données à écrire
arwDataWord[0]	WORD	Mot de données 0 à écrire
arwDataWord[...]	WORD	Mot de données ... à écrire
arwDataWord[n]	WORD	Mot de données n à écrire

Tableau 17

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	Nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	Données lues du tag ...
arTag[n]	DTE810stTagData	Données lues du tag n

Tableau 18

### 6.3.7 ID fonction 7 - Ecriture sur une zone de la mémoire USER d'un tag

Cette fonction écrit une zone à indiquer de la mémoire USER sur un tag défini. Le tag à écrire est sélectionné via l'EPC. Le mot de passe du tag est aussi à saisir. Pour le tag, le mot de passe 0x0000 0000 est réglé par défaut.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (méorisé sur le tag)
stSelectionEPC	DTE810stTagEPCData	Données EPC pour la sélection d'un tag
iLength	INT	Nombre des mots de données EPC du tag sélectionné
arwEpcWord[0]	WORD	Mot de données EPC 0 du tag sélectionné
arwEpcWord[...]	WORD	Mot de données EPC ... du tag sélectionné
arwEpcWord[n]	WORD	Mot de données EPC n du tag sélectionné
stUserDataToWrite	Dte810WriteUserData	Paramètre pour écrire des données de la mémoire USER
diStartAddress	DINT	Adresse mémoire à partir de laquelle l'écriture doit s'effectuer
iLength	INT	Nombre des mots de données à écrire
arwDataWord[0]	WORD	Mot de données 0 à écrire
arwDataWord[...]	WORD	Mot de données ... à écrire
arwDataWord[n]	WORD	Mot de données n à écrire

Tableau 19

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	0 .. nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tableau 20

### 6.3.8 ID fonction 8 – Ecriture d'un EPC sur un tag inconnu

Cette fonction écrit l'EPC indiqué sur un tag. Il ne doit pas être plus d'un tag dans le champ d'antenne. Le mot de passe du tag est aussi à indiquer. Pour le tag, le mot de passe 0x0000 0000 est réglé par défaut.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (mémoire sur le tag)
stEpcToWrite	Dte810TagEpcData	Données EPC à écrire
iLength	INT	Nombre des mots de données EPC à écrire
arwEpcWord[0]	WORD	Mot de données EPC 0 à écrire
arwEpcWord[...]	WORD	Mot de données EPC ... à écrire
arwEpcWord[n]	WORD	Mot de données EPC n à écrire

Tableau 21

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	0 .. nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues avant la modification
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tableau 22

### 6.3.9 ID fonction 9 - Modifier l'EPC d'un tag

Cette fonction écrit l'EPC indiqué sur un tag défini. Le tag à écrire est sélectionné via l'EPC. Le mot de passe du tag est aussi à saisir. Pour le tag, le mot de passe 0x0000 0000 est réglé par défaut.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
dwPassword	DWORD	Mot de passe accès au tag (mémoire sur le tag)
stSelectionEPC	DTE810stTagEPCData	Données EPC pour la sélection d'un tag
iLength	INT	Nombre des mots de données EPC du tag sélectionné
arwEpcWord[0]	WORD	Mot de données EPC 0 du tag sélectionné
arwEpcWord[...]	WORD	Mot de données EPC ... du tag sélectionné
arwEpcWord[n]	WORD	Mot de données EPC n du tag sélectionné
stEpcToWrite	Dte810TagEpcData	Données EPC à écrire
iLength	INT	Nombre des mots de données EPC à écrire
arwEpcWord[0]	WORD	Mot de données EPC 0 à écrire
arwEpcWord[...]	WORD	Mot de données EPC ... à écrire
arwEpcWord[n]	WORD	Mot de données EPC n à écrire

Tableau 23

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	0 .. nombre des tags traités
arTag[1]	DTE810stTagData	Données lues du tag 1
bDataErrorCode	BYTE	Code d'erreur des données de tag (cf. Tableau 29)
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEPC	DTE810stTagEPCData	Données EPC lues avant la modification
iLength	INT	Nombre des mots de données EPC
arwEpcWord[0]	WORD	Mot de données EPC 0 lu
arwEpcWord[...]	WORD	Mot de données EPC ... lu
arwEpcWord[n]	WORD	Mot de données EPC n lu
bTagErrorCode	BYTE	Code d'erreur d'accès au tag (cf. Tableau 30)
stMemoryData	Dte810stTagMemoryData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tableau 24

### 6.3.10 ID fonction 15 - Réglage de la puissance d'antenne

Avec cette fonction la puissance d'antenne peut être réglée. Seulement le port d'antenne raccordé est influencé. S'il y a un raccordement à tous les antennes, le numéro de port d'antenne à influencer est à indiquer.

Paramètres nécessaires dans les variables publiques :

Nom	Type de données	Valeur de saisie
stAntPortPower	Dte810stAntPortPower	Paramètre pour le réglage de la puissance d'antenne
iPortNo	INT	Seulement important si le bloc travaille avec tous les antennes (stSettings.stDTE.iAntPortNo = 0) 1 .. 4      numéro de port à influencer
iPortPower	INT	0      antenne désactivé 68 .. 132      puissance d'antenne en ¼ dBm

Tableau 25

Données de réponse dans les variables publiques :

Nom	Type de données	Valeur de retour
iTagCounter	INT	0
arTag[1]	DTE810stTagData	0x0000
arTag[...]	DTE810stTagData	0x0000
arTag[n]	DTE810stTagData	0x0000

Tableau 26

#### 6.4 Déconnexion et reconnexion de la communication avec le lecteur RFID

Le bloc fonctionnel détecte une déconnexion du lecteur RFID de 2 manières :

Entrée *ixConnOK* = FALSE

Etat Ethernet/IP indique une erreur pendant l'échange de données

Sortie *oxBusy* est TRUE. Une déconnexion pendant l'exécution d'une fonction interrompt la fonction et met la sortie *oxError* = TRUE. Un code d'erreur est retourné dans le paramètre *owStatus*.

Si l'entrée *ixConnOK* = TRUE, la connexion de la communication démarre comme décrit dans le chapitre 6.2.1.

Après terminaison la sortie *oxBusy* = FALSE est fournie.

La séquence des signaux est indiquée dans la Figure 17.

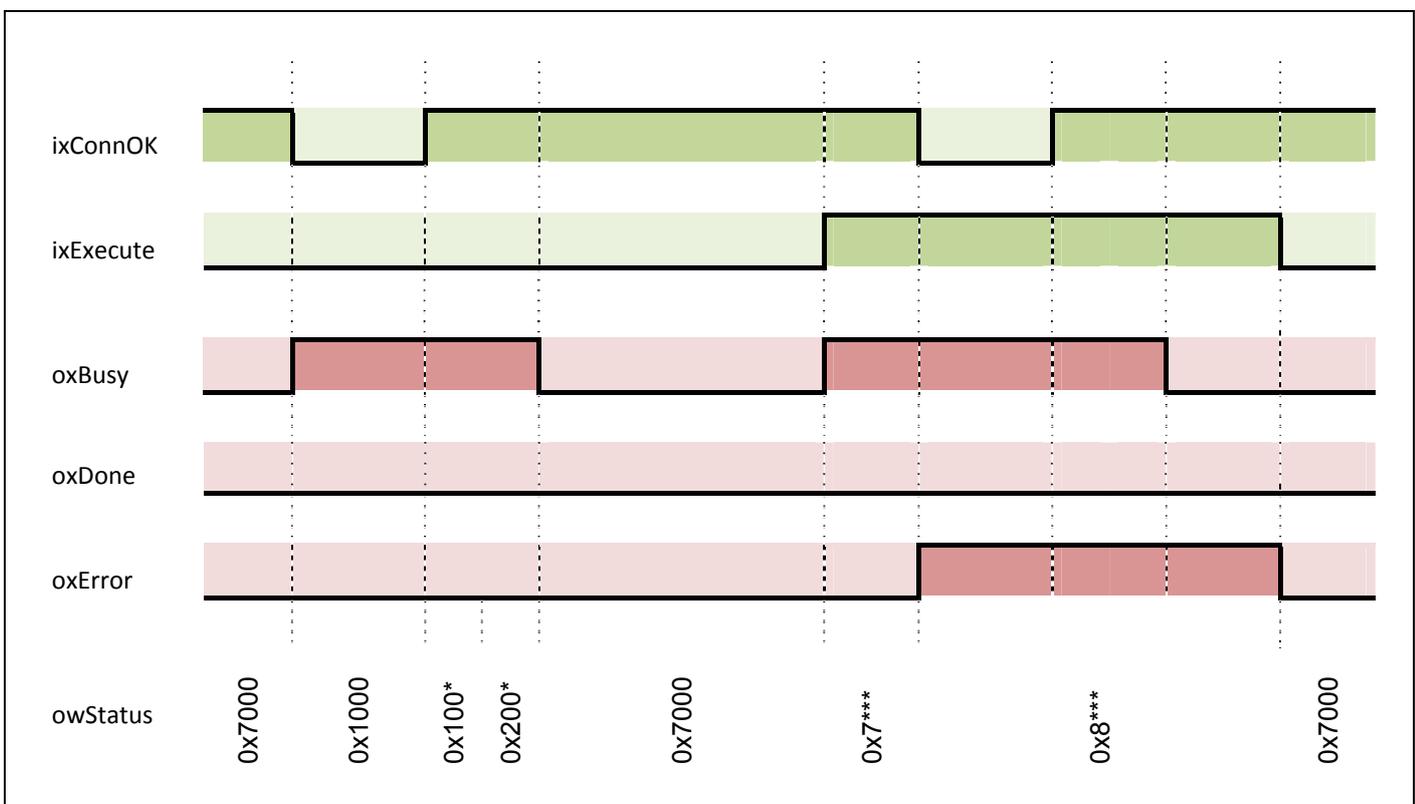


Figure 17

Remarque importante : Le bloc fonctionnel détecte seulement la connexion au lecteur RFID. Le raccordement correct à l'antenne n'est pas surveillé.

## 6.5 Indications d'état sur l'entrée owStatus

owStatus	Description
0x0000	Fonction terminée sans erreur
0x1000 0x1001 0x1002 0x1003 0x1004 0x1005	Entrée ixConnOK = FALSE ou aucun raccordement au lecteur RFID Envoyer la commande de synchronisation Activer la commande de synchronisation Demander les données de réponse Attendre des données, envoyer un accusé de réception Activer l'accusé de réception
0x2000 0x2001 0x2002 0x2003 0x2004 0x2005	Créer une commande d'initialisation Envoyer la commande d'initialisation Activer la commande d'initialisation Demander les données de réponse Attendre des données, envoyer un accusé de réception Activer l'accusé de réception
0x7000 0x7001 0x7002 0x7ab3 0x7ab4 0x7ab5	Attendre le démarrage d'une fonction Envoyer la commande Activer la commande Demander les données de réponse Attendre des données, envoyer un accusé de réception Activer l'accusé de réception
	a = ID fonction active b = 0 attendre des données à évaluer b = 1 vérifier l'attribution des données de réponse b = 2 compter les tags détectés b = 3 transmettre les données de tag à l'array arTag dans les variables publiques
0x8***	Code d'erreur (cf. Tableau 28)

Tableau 27

## 7 Codes d'erreur

### 7.1 Codes d'erreur sur l'entrée owStatus

owStatus	Description
0x8001	Le lecteur RFID retourne NODATA (mode directe : aucune réponse du tag pour les fonctions Gen2)
0x8002	Le lecteur RFID retourne CRCERROR (mode directe : fonctions Gen2 - comparaison de la somme de contrôle : tag faux)
0x8003	Le lecteur RFID retourne NOLICENSE (exécution d'une fonction qui ne tombe pas sous la clé de licence)
0x8004	Le lecteur RFID retourne OUTFRANGE (plage de valeur fausse)
0x8005	Le lecteur RFID retourne NOSTANDARD (mode directe : aucun standard de communication n'a été sélectionné)
0x8006	Le lecteur RFID retourne NOANTENNA (mode directe : aucune antenne activée)
0x8007	Le lecteur RFID retourne NOFREQUENCY (mode directe : aucune fréquence sélectionnée)
0x8008	Le lecteur RFID retourne NOCARRIER (mode directe : support non activé)
0x8009	Le lecteur RFID retourne ANTENNAERROR (mode directe : erreur d'antenne)
0x800A	Le lecteur RFID retourne NOTAG (aucun tag dans le champ ou aucun tag avec l'EPC indiqué dans le champ)
0x800B	Le lecteur RFID retourne MORETHANONETAGINFIELD (plus d'un tag dans le champ, par ex. pour SyncWriteEpcToSingleTag)
0x800C	Le lecteur RFID retourne WRONGLICENSEKEY (clé de licence fausse)
0x800D	Le lecteur RFID retourne FWREJECTED (firmware refusé lorsqu'on essaie de charger un firmware non conforme dans le lecteur)
0x800E	Le lecteur RFID retourne WRONGCFM mode faux (si une commande de mode direct est exécutée en mode normal et vice versa)
0x800F	Le lecteur RFID retourne NOHANDLE (si le handle (authentification 16 bits) du tag ne répond pas)
0x8010	Le lecteur RFID retourne NOPROFILE (mode directe : aucun profil de communication activé)
0x8080	Le lecteur RFID retourne NONSPECIFIED (erreur non spécifiée)
0x8101	Port d'antenne non valable (stSettings.stDTE.iAntPortNo)
0x8102	Taille de l'assemblage non valable réglée dans le lecteur RFID
0x8201	Entrée ixConnOK = FALSE lorsqu'une fonction est active
0x8202	Erreur de la fonction de communication interne DATA_EXCH
0x8203	Ethernet/IP erreur du système
0x8204	Ethernet/IP erreur de protocole
0x8301	Démarrage d'une fonction lorsqu'une fonction est active (ixExecute)
0x8302	ID fonction non valable (iiFunctionID)
0x8303	Longueur de l'EPC sélection non valable (stSelectionEpc.iLength)
0x8304	Longueur de l'EPC à écrire non valable (stEpcToWrite.iLength)
0x8305	Longueur des données de mémoire USER à écrire non valable(stUserDataToWrite.iLength)
0x8401	Début des données de réponse reçues absent
0x8402	Les données de réponse n'appartiennent pas à la commande envoyée
0x8403	L'EPC du tag dans le champ d'antenne trop long
0x8404	Trop de tags dans le champ d'antenne
0x8405	Erreur dans les données d'un tag → Evaluer les codes d'erreur des données de tag (cf. Tableau 29)

Tableau 28

## 7.2 Codes d'erreur des données de tag

arTags[x].bDataErrorCode	Description
0x00	Aucune erreur dans les données de tag
0x80	Erreur lors de l'accès au tag, évaluer le code d'erreur d'accès au tag (cf. Tableau 30)
0x81	Données EPC lues trop longues
0x82	Données de mémoire TID ou USER trop longues

Tableau 29

## 7.3 Codes d'erreur d'accès au tag

arTags[x].bTagErrorCode	Description
0x00	Aucune erreur d'accès au tag
0x01	Aucune réponse du tag
0x02	Accès au tag refusé (mot de passe)
0x03	Comparaison non réussie après l'écriture
0x04	Erreur indéfinie
0x80	Toutes les erreurs qui ne sont pas décrites par d'autres codes d'erreur
0x83	Mémoire tag est plus petites que l'adressage propose ou zone adressée n'existe pas
0x84	Mémoire du tag bloquée (mot de passe) - aucune lecture/écriture possible
0x8B	Puissance insuffisante (pour l'écriture du tag)
0x8F	Tag ne supporte pas les codes d'erreur spécifiés

Tableau 30

## 8 Description des types de données dérivés

### 8.1 Arrays

Nom	Type	Description
Dte810arstTags	ARRAY[1..n] OF Dte810stTagData	Array pour la réception des données des tags traités L'array doit commencer par l'index 1. L'index final n définit combien de tags max peuvent être traités par le DFB. Il doit être configuré pour être compatible avec l'application.
Dte810arwTagEpc	ARRAY[0..n] OF WORD	Array pour la réception de l'EPC d'un tag L'array doit commencer par l'index 0. L'index final n définit la longueur maximale d'un EPC que le DFB peut traiter. Il doit être configuré pour être compatible avec l'application.
Dte810arwTagMemory	ARRAY[0..n] OF WORD	Array pour la réception des données mémoire TID ou USER d'un tag L'array doit commencer par l'index 0. L'index final n définit la taille mémoire maximale que le DFB peut traiter. Il doit être configuré pour être compatible avec l'application.

Tableau 31

## 8.2 Structures

<b>Dte810stAntPortPower</b>		<b>Données pour régler la puissance d'antenne</b>
iPortNo	INT	Numéro de port à influencer
iPortPower	INT	Puissance d'antenne en ¼ dBm

Tableau 32

<b>Dte810stDiagData</b>		<b>Informations sur l'erreur détaillées</b>
iLastFailedFctId	INT	ID fonction avec laquelle la dernière erreur s'est produite
wLastFailedCmdId	WORD	ID commande DTE avec laquelle la dernière erreur s'est produite
wLastErrorCode	WORD	Dernier code d'erreur indiqué par le bloc fonctionnel
wEthipEventCode	WORD	Dernier code d'événement retourné par Ethernet/IP
bEthipCipErrorCode	BYTE	Dernier code d'erreur CIP retourné par Ethernet/IP
wDataExchErrCode	WORD	Dernier code d'erreur retourné par la fonction interne DATA_EXCH

Tableau 33

<b>Dte810stDteSet</b>		<b>Réglages de la connexion au lecteur RFID</b>
ariIPAddr	ARRAY[1..4] OF INT	Adresse IP du lecteur RFID
iAntPortNo	INT	Port d'antenne
iTimeOutVal	INT	Temps de contrôle de la fonction interne DATA_EXCH
iRepeatTime	INT	Temporisation pour répéter la commande quand le lecteur RFID ne fournit pas de données

Tableau 34

<b>Dte810stNocSet</b>		<b>Localisation du module de communication NOC0401 dans l'automate</b>
iRackNo	INT	Numéro du rack dans lequel le module de communication est monté
iSlotNo	INT	Numéro du slot dans lequel le module de communication est monté
iChannelNo	INT	Numéro de la voie du module de communication projetée pour l'échange de données

Tableau 35

<b>Dte810stReadEpc</b>		<b>Paramètre pour la lecture de tous les EPC</b>
iReadingTime	INT	Temps d'activation (en ms) pour la lecture des EPC de tous les tags

Tableau 36

<b>Dte810stReadUserData</b>		<b>Paramètre pour la lecture de la mémoire USER</b>
diStartAddress	DINT	Adresse mémoire à partir de laquelle la lecture doit s'effectuer
iLength	INT	Nombre de mots de données à lire (1 mot = 16 bits)

Tableau 37

<b>Dte810stSettings</b>		<b>Données pour établir une connexion au lecteur RFID</b>
stNOC	Dte810stNocSet	Données pour localiser le module de communication (cf. Tableau 35)
stDTE	Dte810stDteSet	Paramètre pour l'échange de données avec le lecteur RFID (cf. Tableau 34)

Tableau 38

<b>Dte810stTagData</b>		<b>Données lues par un tag</b>
bDataErrorCode	BYTE	Code d'erreur des données du tag
bRssi	BYTE	Valeur RSSI (intensité du signal entre le tag et l'antenne)
stEpc	Dte810stTagEpcData	Données EPC (cf. Tableau 40)
bTagErrorCode	BYTE	Code d'erreur d'accès au tag
stMemoryData	Dte810stTagMemoryData	Données de tags lues (cf. Tableau 41)

Tableau 39

<b>Dte810stTagEpcData</b>		<b>Données d'un EPC</b>
iLength	INT	Nombre de mots de données EPC (1 mot = 16 bits)
arwEpcWord	Dte810arwTagEpc	Array avec mots de données EPC (cf. Tableau 31)

Tableau 40

<b>Dte810stTagMemoryData</b>		<b>Données TID ou USER</b>
iLength	INT	Nombre de mots de données (1 mot = 16 bits)
arwDataWord	Dte810arwTagMemory	Array avec des mots de données (cf. Tableau 31)

Tableau 41

<b>Dte810stWriteUserData</b>		<b>Données pour l'écriture de la mémoire USER d'un tag</b>
diStartAddress	DINT	Adresse mémoire à partir de laquelle l'écriture doit s'effectuer
iLength	INT	Nombre de mots de données à écrire (1 mot = 16 bits)
arwDataWord	Dte810arwTagMemory	Array avec les données à écrire (cf. Tableau 31)

Tableau 42