Protocole de communication

RS232 (5AA)

Disponible pour firmwares :
URM – V10
URF – V10
URL – V05
URI – V02

Table des matières

I. Introduction

Ce document a pour but de permettre la mise en œuvre des produits *STid* utilisant les protocoles de communication ***5AA*** (liaison série RS232).

Ce document décrit les commandes disponibles des lecteurs permettant l'exploitation des puces *EPC1 Gen2.*

II. Interface série

Vitesse de transmission	115200 bauds
Mode	Asynchrone
Nombre de bits	8
Mode de transmission	LSb first
Bit de stop	1

III. Structure du message

3.1. Empaquetage

#02	***Len***	***CTRL***			***Commande***	***CRC***
Start Of Frame (SOF) 0x02	**Longueur de « Commande »**	**@**		**Mode**	**Commande à transmettre**	**CCITT 16 bits CRC**
		Adr	**232/ 485**			
1 octet	**2 octets**	**7 bits**	**1 bits**	**8 bits**	**N octets**	**2 octets**

- # 02 ➔ Marqueur de début de trame Start Of Frame « SOF » (sur un octet 02h).
- Len ➔ Détermine la longueur de la commande à envoyer (deux octets)
- CTRL ➔ Mot de deux octets englobant un octet définissant le mode de communication (message en clair, chiffré, signé etc.…) et un octet définissant le type de liaison série utilisée (RS485 ou RS232).
  - CTRL Mode ➔ Détermine le mode de communication (un octet).
    - 00h ➔ Mode non sécurisé message transmis en clair.
    - 04h ➔ Mode Réservé
  - CTRL @ ➔ Détermine le type de liaison série utilisée (RS232 ou RS485) (bit 0) ainsi que l'adresse du lecteur si liaison RS485 (bit 7 à bit 1).

b7 – b1	***b0***
Adresse du lecteur RS485	***Liaison série utilisée***

    - b0 ➔ ***« 0 »*** RS232 | ***« 1 »*** RS485
    - b7 – b1 ➔ ***« 1111 111 »*** à ***« 0000 000 »***

- CRC ➔ CRC-16-CCITT [Len….Commande] ***[Polynôme « $x^{16} + x^{12} + x^5 + 1$ »*** 0x1021]; *Valeur Initiale 0xFFFF*

Exemple de protocoles utilisant ce CRC ➔ X.25, V.41, CDMA, Bluetooth, PPP, IrDA, BACnet

3.2. *Commande non sécurisée*

- **Commande Host Non sécurisée**

CMD			*Reserved*	L_{out}	$Data_{out}$
RFU	**Type**	**Code**	**AAh 55h**	**Longueur des données que le host envoi**	**Données envoyées par le host**
1 octet	**1 octet**	**2 octets**	**2 octets**	**2 octets**	**L_{out} octets**

En mode non sécurisé, l'indicateur CTRL Mode est égal à 00h 00h et la partie « Commande » est:

- CMD ➔ Mot de quatre octets englobant deux octets déterminant le type de commande (*EPC1 Gen2*) et deux octets définissant le code de la commande à transmettre.
  - Type ➔ Détermine le type de commande (deux octets)
    - 00h 00h ➔ Commande lecteur
    - 00h 08h ➔ Commande *EPC1 Gen2*
  - Code ➔ Détermine le code commande à transmettre au lecteur (deux octets)
- Reserved ➔ AAh 55h (deux octets).
- L_{out} ➔ Détermine la taille des données envoyées par le host (deux octets).
- $Data_{out}$ ➔ Représente les données envoyées par le host (dans le cas d'une écriture par exemple) (L_{out} octets).

- **Réponse lecteur non sécurisée**

ACK	L_{in}	$Data_{in}$	*Status*	
Code commande	**Longueur des données que le host va recevoir**	**Données envoyées par le lecteur**	**Type**	**Code**
2 octets	**2 octets**	**L_{in} octets**	**1 octet**	**1 octet**

- ACK ➔ Acquittement de début de trame, égal au code commande envoyé par le host.
- L_{in} ➔ Détermine la taille des données que le host va recevoir (deux octets).
- $Data_{in}$ ➔ Données envoyées par le lecteur en réponse à la commande du host (L_{in} octets).
- Status ➔ Mot de deux octets représentant le type de statut (lecteur, *EPC1 Gen2*) et le code de résultat de la commande.
  - RFU ➔ 00h
  - Type ➔ Détermine le type de commande (un octet)
    - 00h ➔ Commande lecteur
    - 08h ➔ Commande puce RFid
  - Code ➔ Détermine le code de la commande à transmettre au lecteur (un octet).

- **Trame complète envoyée par le host**

#02	*Len*	*CTRL*	*CMD*	*Reserved*	L_{out}	$Data_{out}$	*CRC*
1 octet	**2 octets**	**2 octets**	**4 octets**	**2 octets**	**2 octets**	**L_{out} octets**	**2 octets**

- **Trame complète envoyée par le lecteur**

#02	*Len*	*CTRL*	*ACK*	L_{in}	$Data_{in}$	*Status*	*CRC*
1 octet	**2 octets**	**2 octets**	**2 octets**	**2 octets**	**L_{in} octets**	**2 octets**	**2 octets**

IV. Commandes

4.1. Lecteur

Pour ce type de commande, l'octet ***Type*** *qui précède le* ***Code commande*** *est à 00h.*

Set_RFSettings

Description

Cette commande permet de paramétrer les cycles de lecture effectués lors des commandes ***Inventory***. Les paramètres ne sont pas sauvegardés en EEProm, ceux-ci seront réinitialisés lorsque le lecteur redémarrera.

Format

Host: 00h 00h 00h 21h AAh 55H 00h 40h A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15}

Lecteur: 00h 21h 00h 00h 00h 00h

Paramètres

Code commande : 00h 21h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 40h
2 Octets

A_n :
4 Octets

Mot de 4 octets déterminant la configuration d'un port logique. Celui est défini par sa durée d'activation lors d'une commande ***Inventory***, de sa puissance et du numéro d'antenne qu'il représente. **[$A_0 \leq A_n \leq A_{15}$]**

2 Octets	*b15 - b4*	*b3 - b0*
ScanDelay	**PowerAnt**	**AntNb**

- **ScanDelay** mot de 2 octets spécifiant la durée d'activation du port logique lors d'une commande ***Inventory*** par tranche de 10 ms.
  [00h 00h ≤ ScanDelay ≤ FFh FFh] soit **[0 ms ≤ ScanDelay ≤ 655350 ms]**
  Si **ScanDelay =** 00h 00h, le port logique sera alors désactivé.

- **PowerAntNb** mot de 2 octets spécifiant la puissance et le numéro d'antenne physique attribuée au port logique.
  Les 12 bits de poids fort (**b15 – b4**) déterminent la puissance utilisée (**PowerAnt**) sur l'antenne.
  Les 4 bits de poids faible (**b3 – b0**) représentent le numéro de l'antenne physique associée au port logique A_n. **[0 ≤ AntNb ≤ 3]**
  [0E6h ≤ PowerAnt ≤ 136h] soit **[23 dBm ≤ PowerAnt ≤ 31 dBm]** ➔ ***URF***
  [003h ≤ PowerAnt ≤ 082h] soit **[3 dBm ≤ PowerAnt ≤ 13 dBm]** ➔ ***URL***

Remarques

Il est nécessaire de transmettre les 64 octets de paramétrages des ports logiques même si ceux-ci ne sont pas exploités. Afin de désactiver un port logique, il est nécessaire de forcer le paramètre **ScanDelay** à 00h 00h (les paramètres **PowerAnt** et **AntNb** seront donc ignorés).

Une antenne physique peut être attribuée sur plusieurs ports logiques avec des délais d'activation et des puissances différents.

Les ports logiques sont attribués par ordre croissant dans la trame ***SetRFSettings*** (**A_0, A_1, …A_n et A_{15}**).

Si plusieurs ports logiques sont actifs, le lecteur les interrogera par ordre croissant (**A_0, A_1, …A_n et A_{15}**).

Par défaut la durée **ScanDelay** d'une commande ***Inventory*** d'un module *URL* est de 100 ms, d'un module *URM* est de 500 ms et de 650 ms sur un module *URF*. Seul le port logique **A_0** est actif avec l'antenne 0 associée (puissance émise sur le module *URF* **30 dBm** soit **12Ch** et **13 dBm** soit **082h** pour le module *URL*).

Dans le cadre d'une utilisation avec un module *URM,* le paramètre de puissance RF ne peut être modifié. La puissance émise sur l'antenne est donc fixe. Dans ce cas, un octet et demi du paramètre **PowerAnt** n'est pas pris en compte

Exemple :

Module URM :
*00h 00h 00h 21h AAh 55h 00h 40h **01h 00h 10h 40h** 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h*

*Dans ce cas de figure, seul le port logique **A_0** est configuré (en bleu souligné) avec comme paramètre d'activation lors d'une commande **Inventory** de 2560 ms. La puissance émise étant fixe et le module n'ayant qu'une seule antenne (N°0), le paramètre **PowerAnt** n'est donc pas pris en compte.*

Note :

Dans le cadre d'une utilisation d'un module *URM*, il n'est pas nécessaire de configurer les seize ports logiques hormis **A_0**. La configuration des ports **A_1 à A_{15}** peut être utilisée en revanche pour établir des temps d'***Inventory*** différents.

Exemple :

Module URF :

00h 00h 00h 21h AAh 55H 00h 40h 01h 00h 0Eh 60h 01h 00h 11h 01h 50h 00h 0Fh 82h 00h 0Ah 1Bh 13h 0Ah 0Bh 12h 00h 99h 00h 0Dh 01h 01h 30h 0Fh 12h 09h 76h 0Eh 83h 00h 3Fh 0Fh 50h 0Ah 02h 11h 10h 04h 00h 12h A1h D0h 21h 0Eh 01h 99h 00h 1Ah 02h 00h 04h 0Eh 62h 00h 80h 10h 03h 00h FFh 0Eh 93h

$ScanDelay_{A0}$	:	*01h 00h*
$PowerAntNb_{A0}$	:	*0Eh 60h*
$ScanDelay_{A1}$	:	*01h 00h*
$PowerAntNb_{A1}$	:	*11h 01h*
$ScanDelay_{A2}$	:	*50h 00h*
$PowerAntNb_{A2}$	:	*0Fh 82h*
$ScanDelay_{A3}$	:	*01h 0Ah*
$PowerAntNb_{A3}$	:	*1Bh 13h*
$ScanDelay_{A4}$	:	*0Ah 0Bh*
$PowerAntNb_{A4}$	:	*12h 00h*
$ScanDelay_{A5}$	:	*00h 99h*
$PowerAntNb_{A5}$	:	*0Dh 01h*
$ScanDelay_{A6}$	:	*01h 30h*
$PowerAntNb_{A6}$	:	*0Fh 12h*
$ScanDelay_{A7}$	:	*09h 76h*
$PowerAntNb_{A7}$	:	*0Eh 83h*
$ScanDelay_{A8}$	:	*00h 3Fh*
$PowerAntNb_{A8}$	:	*0Fh 50h*
$ScanDelay_{A9}$	:	*0Ah 02h*
$PowerAntNb_{A9}$	:	*11h 10h*
$ScanDelay_{A10}$	:	*04h 00h*
$PowerAntNb_{A10}$	:	*12h A1h*

ScanDelay$_{A11}$: *D0h 21h*
PowerAntNb$_{A11}$: *0Eh 01h*

ScanDelay$_{A12}$: *99h 00h*
PowerAntNb$_{A12}$: *1Ah 02h*

ScanDelay$_{A13}$: *00h 04h*
PowerAntNb$_{A13}$: *0Eh 62h*

ScanDelay$_{A14}$: *00h 80h*
PowerAntNb$_{A14}$: *10h 03h*

ScanDelay$_{A15}$: *00h FFh*
PowerAntNb$_{A15}$: *0Eh 93h*

Reset_RFSettings

Description

Cette commande permet de réinitialiser les paramètres par défauts des cycles de lecture effectués lors des commandes ***Inventory***. Ces informations seront sauvegardées en EEProm.

Format

Host: 00h 00h 00h 23h AAh 55H 00h 00h

Lecteur: 00h 23h 00h 00h 00h 00h

Paramètres

Code commande : 00h 23h
2 Octets

L$_{in}$: 00h 00h
2 Octets

L$_{out}$: 00h 00h
2 Octets

Remarques

Il est nécessaire de redémarrer le lecteur afin que cette commande soit prise en compte.

Set_RFSettings_Saved

Description

Cette commande permet de paramétrer les cycles de lecture effectués lors des commandes ***Inventory***. Ces informations seront sauvegardées en EEProm.

Format

Host: 00h 00h 00h 22h AAh 55H 00h 40h A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15}

Lecteur: 00h 22h 00h 00h 00h 00h

Paramètres

Code commande : 00h 22h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 40h
2 Octets

A_n : Mot de 4 octets déterminant la configuration d'un port logique. Celui est défini par sa durée d'activation lors d'une commande ***Inventory***, de sa puissance et du numéro d'antenne qu'il représente. **[$A_0 \leq A_n \leq A_{15}$]**
4 Octets

2 Octets	*b15 - b4*	*b3 - b0*
ScanDelay	**PowerAnt**	**AntNb**

- **ScanDelay** mot de 2 octets spécifiant la durée d'activation du port logique lors d'une commande ***Inventory*** par tranche de 10 ms.
  [00h 00h ≤ ScanDelay ≤ FFh FFh] soit **[0 ms ≤ ScanDelay ≤ 655350 ms]**
  Si **ScanDelay =** 00h 00h, le port logique sera alors désactivé.

- **PowerAntNb** mot de 2 octets spécifiant la puissance et le numéro d'antenne physique attribuée au port logique.
  Les 12 bits de poids fort (**b15 – b4**) déterminent la puissance utilisée (**PowerAnt**) sur l'antenne. **[0E6h ≤ PowerAnt ≤ 12Ch]** soit **[20 dBm ≤ PowerAnt ≤ 30 dBm]**
  Les 4 bits de poids faible (**b3 – b0**) représentent le numéro de l'antenne physique associée au port logique **A_n**. **[0 ≤ AntNb ≤ 3]**

Remarques

Cette fonction est identique à ***SetRFSettings***. Seul le code commande change (CF commande ***SetRFSettings*** pages 7-8).

Get_RFSettings

Description

Cette commande permet de récupérer les paramètres de configuration RF des seize ports logiques du lecteur.

Format

Host: 00h 00h 00h 20h AAh 55h 00h 00h

Lecteur: 00h 20h 00h 40h A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} 00h 00h

Paramètres

Code commande : 00h 20h
2 Octets

L_{in}: 00h 40h
2 Octets

L_{out}: 00h 00h
2 Octets

A_n : Mot de 4 octets déterminant la configuration d'un port logique. Celui est défini par sa durée d'activation lors d'une commande ***Inventory***, de sa puissance et du numéro d'antenne qu'il représente. **[$A_0 \leq A_n \leq A_{15}$]**
4 Octets

2 Octets	*b15 - b4*	*b3 - b0*
ScanDelay	PowerAnt	AntNb

- **ScanDelay** mot de 2 octets spécifiant la durée d'activation du port logique lors d'une commande ***Inventory*** par tranche de 10 ms.
  [00h 00h ≤ ScanDelay ≤ FFh FFh] soit **[0 ms ≤ ScanDelay ≤ 655350 ms]**
  Si **ScanDelay =** 00h 00h, le port logique est alors désactivé.

- **PowerAntNb** mot de 2 octets spécifiant la puissance et le numéro d'antenne physique attribuée au port logique.
  Les 12 bits de poids fort (**b15 – b4**) déterminent la puissance utilisée (**PowerAnt**) sur l'antenne. **[0E6h ≤ PowerAnt ≤ 12Ch]** soit **[20 dBm ≤ PowerAnt ≤ 30 dBm]**
  Les 4 bits de poids faible (**b3 – b0**) représentent le numéro de l'antenne physique associée au port logique **A_n**. **[0 ≤ AntNb ≤ 3]**

Exemple :

Module URF :

00h 00h 00h 20h AAh 55h 00h 40h 00h 04h 0Eh 62h 96h 00h 11h 03h 5Ah 28h 12h C0h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 08h 00h

$ScanDelay_{A0}$: *00h 04h*
$PowerAntNb_{A0}$: *0Eh 62h*

$ScanDelay_{A1}$: *96h 00h*
$PowerAntNb_{A1}$: *11h 03h*

$ScanDelay_{A2}$: *5Ah 28h*
$PowerAntNb_{A2}$: *12h C0h*

Get_HealthParameters

Description

Cette commande permet de récupérer les paramètres vitaux du module *URF*.

Format

Host: 00h 00h 00h 24h AAh 55h 00h 00h

Lecteur: 00h 24h 00h 06h $Tune_0$ $Tune_1$ $Tune_2$ $Tune_3$ TempCore TempPA 00h 00h

Paramètres

Code commande : 00h 24h
2 Octets

L_{in}: 00h 06h
2 Octets

L_{out}: 00h 00h
2 Octets

$Tune_{0-3}$: Octet représentant l'état actuel de l'accord de l'antenne physique.
1 Octet $Tune_0$ correspondant à l'antenne 0 jusqu'à $Tune_3$ pour l'antenne 3.

- 00h ≤ $Tune_{0-3}$ ≤ 64h : Mauvaise antenne ou antenne non connectée.
- 64h ≤ $Tune_{0-3}$ ≤ A0h : Antenne ou connexion médiocre.
- A0h ≤ $Tune_{0-3}$ ≤ FFh : Antenne correctement connectée et de bonne qualité.

TempCore: Octet représentant la température du cœur du lecteur en °C.
1 Octet

TempPA: Octet représentant la température de l'amplificateur de puissance du lecteur en °C.
1 Octet

Remarques

L'octet $Tune_{0-3}$ est une image du *ROS* de l'antenne.

Cette commande n'est pas disponible sur les modules *URM* et *URL*.

Autonomous_Start

Description

Cette commande permet de démarrer le mode autonome du lecteur. Dans ce mode le lecteur ne lira uniquement que les numéros *EPC* et ne répondra qu'à la commande ***Autonomous_Stop***.

Format

Host: 00h 00h 00h 10h AAh 55h 00h 00h

Lecteur: 00h 10h 00h 00h 00h 00h

Paramètres

Code commande : 00h 10h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 00h
2 Octets

Remarques

Le format de la trame est défini par la fonction ***Autonomous_Output.***

Le lecteur redémarrera automatiquement après réception de cette commande.

Autonomous_Stop

Description

Cette commande permet d'arrêter le mode autonome du lecteur. Celle-ci est la seule commande que le lecteur acceptera dans le mode autonome.

Format

Host: 00h 00h 00h 11h AAh 55h 00h 00h

Lecteur: 00h 11h 00h 00h 00h 00h

Paramètres

Code commande : 00h 11h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 00h
2 Octets

Remarques

Le lecteur redémarrera automatiquement après réception de cette commande.

Autonomous_Output

Description

Cette commande permet de configurer le format de la trame du lecteur dans le mode autonome.

Format

Host: 00h 00h 00h 12h AAh 55h 00h 03h OutConf ReadConf Len

Lecteur: 00h 12h 00h 00h 00h 00h

Paramètres

Code commande : 00h 12h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 03h
2 Octets

OutConf: Octet permettant de définir le format de la trame série.
1 Octet

b7	*b6*	*b5*	*b4*	*b3*	*b2*	*b1*	*b0*
Port logique	*NoLeadingZeros*	*ASCII*	*MSB / LSB*	*Hexa / Déc.*	*LRC*	*CR / LF*	*STX / ETX*
0 Pas de port logique remonté par le lecteur *1 Port logique remonté par le lecteur*	*0 NoLeadingZeros désactivé* *1 NoLeadingZeros activé*	*0 désactive le mode ASCII* *1 Active le mode ASCII*	*0 MSB First* *1 LSB First*	*0 Hexadécimal* *1 Décimal*	*0 LRC désactivé* *1 LRC activé*	*0 CR / LF désactivé* *1 CR / LF activé*	*0 STX / ETX désactivé* *1 STX / ETX activé*

ReadConf: Octet permettant de définir le format de la trame série.
1 Octet

b7	*b6*	*b5*	*b4*	*b3*	*b2*	*b1*	*b0*
EPC	*TID*	*RFU*	*RFU*	N_{WORD3}	N_{WORD2}	N_{WORD1}	N_{WORD0}
0 Lecture d'EPC demandée *1 Lecture d'EPC non demandée*	*0 Lecture de TID non demandée* *1 Lecture de TID demandée*	*X*	*X*				

- Bit 7 – EPC : Active / désactive la lecture de l'EPC en mode autonome.
- Bit 6 – TID : Active / désactive la lecture du TID en mode autonome.
- Bit 5 – Bit 4 – RFU : Non utilisé. « 0 » ou « 1 ».
- Bit 3 – Bit 0 $N_{WORD3-0}$: Représente la longueur en mots de deux octets du TID à lire - 1. [0xx0… 0xxF] +1. $N_{WORD3-0max}$ = *F.* $TID_{remonté\ max}$ = (F + 1) x2 soit 32 octets de TID

Len :
1 Octet

Octet représentant la longueur (en octets) de l'*EPC/TID* à recevoir. Si la longueur demandée est supérieure à celle de l'*EPC* courant, le lecteur rajoute alors en poids fort (MSB) des zéros jusqu'à la taille désirée. Inversement si la longueur demandée est inférieure à la longueur de l'*EPC/TID* réellement lu, alors le lecteur tronque les bits de poids fort. Si l'option *NoLeadingZeros* est activée, il se peut que la taille désirée ne soit pas respectée.
[01h … 07h] en décimal et [01h … 0Ah] en hexadécimal.

Remarques

- ✓ Le LRC est calculé sur les informations suivantes : *EPC (XOR) Port Logique*
- ✓ En décimal, la taille est limitée à 7 octets.
- ✓ CR+LF = (0Dh+0Ah)
- ✓ Le mode ASCII impacte uniquement les données *EPC, Port Logique* ainsi que les zéros de données. La longueur des données est alors doublée.
- ✓ L'information *Port Logique* remontée ne concerne que le premier Port Logique ayant lu *l'EPC.*
- ✓ Pour une lecture de l'EPC **et** du TID, le lecteur retournera deux trames. La première pour l'EPC et la seconde pour le TID. Les modalités de sortie sont fixées par l'octet **OutConf**.
- ✓ Il est important de connaître la taille du TID à lire sur les puces EPC1GEN2. Si le paramètre $\mathbf{N_{WORD}}$ est plus grand que la taille du TID à lire, le lecteur retournera l'erreur **08h 02h** correspondant à une erreur EPC1GEN2 (Voir Chapitre 1.5.2 Codes erreurs EPC1 Gen2 pour plus d'informations).

Trame de sortie pour la lecture d'un *EPC et/ou TID*:

STX	*Port logique*	*EPC/TID*	*LRC*	*CR*	*LF*	*ETX*
02h	*1 à 2 octets*	*Len octets*	*1 octet*	*0Dh*	*0Ah*	*03h*

Différents modes possibles :

Mode 1

Utilisé lorsque **ReadConf** = **0b00xxxxxx**.
Dans ce mode, qui ne concerne que le code EPC, la remontée des codes EPC de plusieurs tags dans le champ à l'issus d'un scan est possible. Dans ce cas le délai entre deux codes EPC sur la liaison série est d'environ 1ms.

Mode 2

ReadConf = **0b01xxxxxx**
ReadConf = **0b11xxxxxx**
ReadConf = **0b10xxxxxx**

Second mode utilisé. Dans celui-ci, un seul tag est remonté par séquence RF. Si deux tags sont présents il faudra **au moins** deux séquences pour remonter les codes relatifs à ces deux scans.

Exemple de comportement suivant les modes 1 ou 2 avec un lecteur configuré avec 3 ports logiques:

	Placement des tags	*scan port 0*	*Fin du scan port 0*	*scan port 1*	*Fin du scan port 1*	*scan port 2*	*Fin du scan port 2*
Mode 1	1 tag présent devant chaque port logique.		.....		.....		**3 codes EPC remontés**
	Tag présent sur 0 et 1.		.....		.....		**2 codes EPC remontés**
	2 tags présents devant le port logique 1.		.....		.....		**2 codes EPC remontés**
Mode 2	1 tag présent devant chaque port logique.		**Code(s) du tag 0 remonté**		**Code(s) du tag 1 remonté**		**Code(s) du tag 2 remonté**
	1 tag présent devant les ports logiques 0 et 1 et pas de tag sur le porte logique 2.		**Code(s) du tag 0 remonté**		**Code(s) du tag 1 remonté**		.....
	2 tags présents devant le port logique 1.		**Code(s) de 1 des 2 tags remonté(s)**		.....		.....

SetOptoOutputParam

Description

Cette commande permet d'activer sur les modules *URL* une sortie optocouplée (broche *Output*) qui changera d'état à l'issue de la commande ***Inventory*** ayant détectée un ou plusieurs tags. L'état passera de « 1 » à « 0 ».

Format

Host: 00h 00h 00h 25h AAh 55h 00h 01h Delay

Lecteur: 00h 25h 00h 00h 00h 00h

Paramètres

Code commande : 00h 25h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 01h
2 Octets

Delay : Octet spécifiant la durée de l'état bas (x 10ms). La valeur 00h désactive la sortie.
1 Octet

Remarques

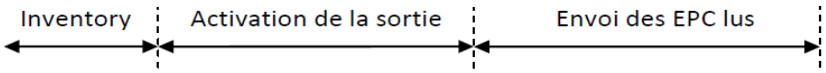
Cette fonction est également disponible en mode autonome. Elle doit être configurée au préalable en mode normal.

Broche *Output* du module *URL* (Pin 6 du connecteur J1 – se reporter à la documentation NI0268A0x).

Cette commande n'est pas disponible sur les modules *URM* et *URF*.

Le module *URL* ne répondra pas aux requêtes durant l'activation de la sortie *Output*. Il est possible que l'utilisation de cette fonction réduise la fluidité des échanges selon les temps d'activation de la sortie et du délai de l'*Inventory*.

Si la sortie est activée, le fonctionnement sera de la forme suivante :



ChangeRegulation

Description

Cette commande permet de changer la régulation du lecteur respectée par le lecteur (fréquences et protocole s de fonctionnement imposés par cette régulation).

Format

Host: 00h 00h 00h 26h AAh 55h 00h 01h Reg

Lecteur: 00h 26h 00h 00h 00h 00h

Paramètres

Code commande : 00h 26h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 01h
2 Octets

Reg : Octet déterminant la régulation et le reset à imposer au lecteur. Déterminé tel que :
1 Octet

- Le quartet de poids fort représente le redémarrage automatique du lecteur.
  - 0000_b Pas de redémarrage du lecteur après réception de la commande.
  - 0001_b Redémarrage du lecteur après réception de la commande.
- Le quartet de poids faible représente la régulation imposée au lecteur.
  - 0000_b FCC
  - 0001_b ETSI

Remarques

Pour prendre en compte une modification de la régulation le lecteur doit être redémarré.

Tous les modules (*URL*, *URM* et *URF*) implémentent cette fonction. Cependant, seul le module URL (référence URL-Wx1-A-U04-5AA ou URL-Wx1-B-U04-5AA) dispose d'une architecture mixte ETSI & FCC. Un module n'étant pas doté de cette architecture donnera des résultats médiocres dans une régulation différente de celle prévue initialement.

Cette fonction écrit des paramètres en EEPROM. Il est donc fortement déconseillé de s'en servir continuellement afin de changer de régulation lors d'échanges RF. Cette commande doit être seulement utilisée pour ajuster la régulation du lecteur à celle en vigueur dans le pays d'installation.

GetInfos

Description

Cette commande permet de récupérer les valeurs des paramètres du lecteur.

Format

Host : 00h 00h 00h 08h AAh 55h 00h 00h

Lecteur : 00h 08h 00h 05h Version Baudrate AddressRS485 Day Month 00h 00h

Paramètres

Code commande : 00h 08h
2 Octets

L_{in}: 00h 05h
2 Octets

L_{out} : 00h 00h
2 Octets

Version: Octet indiquant l'indice software du lecteur.
1 Octet

Baudrate: Octet indiquant la vitesse de communication utilisée par le lecteur.
1 Octet [00h … 04h].

- 00h 9600 bauds
- 01h 19600 bauds
- 02h 38400 bauds
- 03h 57600 bauds
- 04h 115200 bauds

AddressRS485: Octet indiquant l'adresse actuelle du lecteur RS485.
1 Octet

Day : Information codée sur 1 octet représentant le jour de compilation du firmware.
1 Octet

Month: Information codée sur 1 octet représentant le mois de compilation du firmware.
1 Octet

SetBaudRate

Description

Cette commande permet de changer la vitesse de communication série des lecteurs.

Format

Host : 00h 00h 00h 05h AAh 55h 00h 01h Baudrate

Lecteur : 00h 05h 00h 00h 00h 00h

Paramètres

Code commande : 00h 05h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 01h
2 Octets

Baudrate : Octet indiquant la vitesse à laquelle le lecteur doit fonctionner.
1 Octet [00h ; 04h]

- 00h 9600 bauds
- 01h 19600 bauds
- 02h 38400 bauds
- 03h 57600 bauds
- 04h 115200 bauds

Remarques

Si la valeur de l'octet Baudrate est supérieure à 04h, la vitesse de communication sera fixée à 9600 bauds.

Set485Address

Description

Cette commande permet de changer l'adresse RS485 des lecteurs.

Format

Host : 00h 00h 00h 06h AAh 55h 00h 01h Address

Lecteur : 00h 06h 00h 00h 00h 00h

Paramètres

Code commande : 00h 06h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 01h
2 Octets

Adress : Octet déterminant la valeur de l'adresse du lecteur à changer.
1 Octets Valeur comprise entre [0 … 127] soit [00h … 7Fh].

Remarques

Cette commande n'est valable que pour les lecteurs type RS485. Le bit « 0 » de l'octet « CTRL-@ » devra donc être forcé à « 1 ». La valeur 00h permet au lecteur de répondre à toutes les requêtes quelque soit l'adresse définie dans celle-ci. C'est l'adresse de diffusion globale.

Set_RF_Param

Description

Cette commande permet de modifier certains paramètres du protocole ISO18000-6C.

Format

Host : 00h 00h 00h 27h AAh 55h 00h 07h NV/V Address Data

Lecteur : 00h 05h 00h 00h 00h 00h

Paramètres

Code commande : 00h 27h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 07h
2 Octets

NV/V : Octet permettant la sauvegarde en EEPROM ou non de la valeur de **Data**.
1 Octet [00h ; 01h]
- 00h La valeur **Data** sera effacée à l'arrêt du lecteur. Valeur par défaut restaurée.
- 01h La valeur **Data** sera conservée à l'arrêt du lecteur.

Address: 00h 01h. Représente le paramètre *Q*.
2 Octets

- Le paramètre *Q* est un paramètre essentiel pour l'anticollision. Il détermine l'ampleur de la population de tags « détectables » lors de chaque round d'***Inventory*** (un ***Inventory*** étant composé de plusieurs rounds).
- Plus le paramètre *Q* est petit, plus vite sera parcouru l'algorithme d'anticollision. S'il est certain qu'un seul tag sera présent devant l'antenne, il sera efficace de choisir une valeur *Q* = 0 ou 1.

Data: Valeur du paramètre à modifier représenté par la valeur **Address**.
4 Octets *Q* = [00h 00h 00h 00h … 00h 00h 00h 0Fh]. Par défaut 00h 00h 00h 07h.

Remarques

Si pour une application donnée, l'utilisateur connait à priori le nombre de tags présent devant une antenne lors d'un ***Inventory***, celui peut alors adapter l'anticollision afin d'accélérer la vitesse de lecture.

Toute variation d'un paramètre RF par le biais de cette commande en mode non volatile (**NV/N** = 00h) sera conservée après le changement de la régulation par la commande ***ChangeRegulation***.

Attention, si la population de tags utilisés est trop grande pour la valeur du paramètre *Q*, de nombreuses collisions se produiront rendant le processus long voire aveugle pour certains tags de cette population.

Le lecteur utilise le paramètre *Q* de façon différente selon les opérations demandées et l'algorithme choisi. Celui-ci n'est pris en compte que pour les opérations suivantes.

- ***Read***, ***Lock***, ***Kill*** et ***Write***.
- En mode *Autonomous* – Mode 2 uniquement.

Cette valeur ne sera donc **pas** prise en compte pour la commande ***Inventory*** et pour le mode 1 de *l'Autonomous*.

Il est recommandé de se référer à la norme **ISO_IEC_FDIS_18000-6C** pour plus d'information sur l'utilisation du paramètre *Q*.

La valeur *Q* peut être ajustée en fonction de la population de tag avec la formule suivante : $2^Q - 1$.

Certains paramètres modifiant le protocole ISO18000-6C seront implémentés dans le futur.

Retrieve_RF_Params

Description

Cette commande permet de retrouver **tous** les paramètres RF qui ont pu être changés par une ou plusieurs commandes ***Set_RF_Params*** (**NV/V** = 00h). **Tous** ces paramètres retrouvent leur valeur par défaut.

Format

Host : 00h 00h 00h 28h AAh 55h 00h 00h

Lecteur : 00h 05h 00h 00h 00h 00h

Paramètres

Code commande : 00h 28h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: 00h 00h
2 Octets

4.2. EPC1 Gen2

Pour ce type de commande, l'octet ***Type*** *qui précède le* ***Code commande*** *est à 08h.*

Inventory

Description

Cette commande retourne la totalité des puces *EPC1 Gen2* détectées par le lecteur lors d'un cycle de lecture.

Format

Host: 00h 08h 00h 01h AAh 55h 00h 00h

Lecteur: 00h 01h L_{in} NbTags [EPCLen EPC AntID NbRead] ^{nbTags} 08h 00h

Paramètres

Code commande : 00h 01h
2 Octets

L_{in}: Valeur variable en fonction de la longueur des codes EPC1 Gen2 ainsi que du nombre
2 Octets total de tags lus. **NbTags + ([EPCLen + EPC + AntID + NbRead] x NbTags)**

L_{out}: 00h 00h
2 Octets

NbTags: Cet octet indique le nombre d'identifiants *EPC1 Gen2* différents détectés par le lecteur.
1 Octet [00h … F7h]. 247 tags détectés maximum.

EPCLen : Octet déterminant la longueur (en octets) de l'identifiant *EPC1 Gen2* qui suit.
1 Octet Seuls les identifiants de 96 bits seront détectés.

EPC : Identifiant *EPC1 Gen2* du tag lu. Sa taille dépendant de la valeur **EPCLen**.
EPCLen Octets

AntID : Représente le numéro du premier port logique sur lequel l'identifiant *EPC1 Gen2* a
1 Octet été détecté. Par adéquation, un port logique représente une antenne (voir commandes ***SetRFSettings*** et ***GetRFSettings***)

NbRead : Cet octet représente le nombre de lecture d'un tag *EPC1 Gen2* sur la totalité des ports
2 Octets logiques par commande ***Inventory***.

Remarques

Les paramètres **[EPCLen EPC AntID NbRead]** sont transmis autant de fois qu'il y a de tags *EPC1 Gen2* lus (**NbTags**). Si aucun tag n'est détecté, ces paramètres ne seront pas transmis. Seul l'octet **NbTags** sera inclus dans la trame avec pour valeur 00h.

Exemple :

00h 01h 00h 1Fh 02h 0Ch E7h CDh 52h 46h E9h C3h A8h 4Ch 5Dh 32h 61h 86h 01h 0Ah 0Ch BDh 69h 88h 64h 43h 48h D2h EEh 43h 1Eh F4h 13h 0Bh F0h 08h 00h

L_{in}:	1Fh	(31 octets)
NbTags:	02h	*(2 tags lus)*
$EPCLen_1$:	0Ch	*(Id de l'identifiant sur 12 octets)*
EPC_1:	E7h CDh 52h 46h E9h C3h A8h 4Ch 5Dh 32h 61h 86h	
$AntID_1$:	01h	*(Identifiant détecté sur le port logique 1)*
$NbRead_1$:	00h 0Ah	*(Identifiant lu 10 fois)*
EPCLen2:	0Ch	*(Id de l'identifiant sur 12 octets)*
EPC_2:	BDh 69h 88h 64h 43h 48h D2h EEh 43h 1Eh F4h 13h	
AntID2:	0Bh	*(Identifiant détecté sur le port logique 11)*
$NbRead_2$:	00h F0h	(Identifiant lu 240 fois)

Il est également possible d'activer la commande ***Inventory*** en appliquant un état haut sur l'entrée *Input* du module *URL* (Pin 2 du connecteur J1 – se reporter à la documentation NI0268A0x pour l'utilisation de cette fonction).

Kill

Description

Commande permettant de désactiver irrémédiablement le tag identifié en fonction d'un masque (**Mask**).

Format

Host: 00h 08h 00h 04h AAh 55H L_{out} MaskBank MaskLen MaskOffset Mask PWD Recom LogicalPort

Lecteur: 00h 04h 00h 00h 08h 00h

Paramètres

Code commande : 00h 04h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: Valeur variable en fonction de la longueur du masque (**Mask**) [09h + **MaskLen**].
2 Octets

MaskBank: 01h (RFU i.e. Bank EPC).
1 Octet

MaskLen: Longueur du masque EPC en octets [00h ... 1Eh].
1 Octet

MaskOffset: Décalage dans le masque en octets par rapport à la Bank EPC [04h ... 21h].
1 Octet

Mask: Correspond au masque EPC permettant au lecteur de sélectionner le(s) tag(s).
MaskLen Octets

PWD: Mot de 4 octets représentant le mot de passe **Kill Password** MSB first.
4 Octets

Recom : RFU = 00h
1 Octets

LogicalPort: Octet représentant le port logique sur lequel le lecteur tentera de désactiver le tag. [00h ... 0Fh]
1 Octet

Remarques

Cette action est irréversible. Un tag désactivé devient inutilisable.

Le lecteur utilisera les paramètres d'antenne et de puissance associés au port logique en fonction des paramètres par défauts ou définis par les commandes ***Set_RFSettings*** ou ***Set_RFSettings_Saved***. Le paramètre de temps est quant à lui figé.

Si le port logique utilisé dans la commande n'est pas attribué, le lecteur remontera l'erreur **00h 02h**.

Il est nécessaire que le *Kill Password* soit différent de 00h 00h 00h 00h dans la Bank *Reserved* afin de pouvoir utiliser la commande ***Kill***.

Il est important d'indiquer le mot de passe **PWD** (par défaut 00h 00h 00h 00h) même si la Bank n'est pas sécurisée. Dans le cas où celui-ci est différent de 00h 00h 00h 00h et que la Bank n'est pas sécurisée, il est possible d'utiliser la valeur par défaut ou la valeur réelle.

Lock

Description

Cette commande permet de verrouiller en écriture un tag en fonction d'un masque (**Mask**).

Format

Host: 00h 08h 00h 05h AAh 55H L_{out} MaskBank MaskLen MaskOffset Mask PayLoadMask PayLoadAction PWD LogicalPort

Lecteur: 00h 05h 00h 00h 08h 00h

Paramètres

Code commande : 00h 05h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: Valeur variable en fonction de la longueur du masque (**Mask**) [0Ch + **MaskLen**].
2 Octets

MaskBank: 01h (RFU i.e. Bank EPC).
1 Octet

MaskLen: Longueur du masque EPC en octets [00h ... 1Eh].
1 Octet

MaskOffset: Décalage dans le masque en octets par rapport à la Bank EPC [04h ... 21h].
1 Octet

Mask: Correspond au masque EPC permettant au lecteur de sélectionner le(s) tag(s).
MaskLen Octets

PayLoadMask: Mot de 2 octets représentant un masque validant ou invalidant les actions de verrouillage définies par le **PayLoadAction**. Ces deux octets sont calculés de la façon suivante :
2 Octets

b15 - b10	b9 - b8	b7 - b6	b5 - b4	b3 - b2	b1 - b0
000000	Kill	Access	EPC	TID	User

Avec les paramètres suivants:

- $[bit_x - bit_x]$ ➔ OO Bits du **PayLoadAction** correspondants non pris en compte.
- $[bit_x - bit_x]$ ➔ 11 Bits du **PayLoadAction** correspondants pris en compte.

PayLoadAction: Mot de 2 octets définissant les actions de verrouillage sur le tag sélectionné.
2 Octets
Ces deux octets sont calculés de la façon suivante :

b15 - b10	b9 - b8	b7 - b6	b5 - b4	b3 - b2	b1 - b0
000000	Kill	Access	EPC	TID	User

Il est possible de verrouiller de 4 façons les Banks *User*, *TID* et *EPC*.

- $[bit_x - bit_x]$ ➔ 00 ➔ Writeable ➔ La Bank peut être écrite avec/sans mot de passe.
- $[bit_x - bit_x]$ ➔ 01 ➔ AlwaysWriteable ➔ La Bank est toujours Writeable.
- $[bit_x - bit_x]$ ➔ 10 ➔ SecuredWriteable ➔ L'écriture sur la Bank nécessite un mot de passe.
- $[bit_x - bit_x]$ ➔ 11 ➔ AlwaysNOTWriteable ➔La Bank ne peut plus être écrite.

Et 4 autres pour la Bank *Reserved* contenant les mots de passe *Access Password* et *Kill Password*:

- [bit_x – bit_x] ➔ 00 ➔ Accessible ➔ Le mot de passe est accessible (lecture et écriture) avec/sans mot de passe.
- [bit_x – bit_x] ➔ 01 ➔ AlwaysAccessible ➔ Le mot de passe est toujours Accessible.
- [bit_x – bit_x] ➔ 10 ➔ SecuredAccessible ➔ L'accès au mot de passe nécessite un mot de passe.
- [bit_x – bit_x] ➔ 11 ➔ AlwaysNOTAccessible ➔ Le mot de passe n'est plus accessible.

PWD:
4 Octets
Mot de 4 octets représentant le mot de passe **Access Password** MSB first.

LogicalPort:
1 Octet
Octet représentant le port logique sur lequel le lecteur tentera d'effectuer l'opération sur le tag. [00h ... 0Fh]

Remarques

Les actions *AlwaysNOTAccessible, AlwaysAccessible, AlwaysWriteable, AlwaysNOTWriteable* sont irréversibles et ne peuvent être modifiées par une autre commande ***Lock***.

Il est important d'indiquer le mot de passe **PWD** (par défaut 00h 00h 00h 00h) même si la Bank n'est pas sécurisée. Dans le cas où celui-ci est différent de 00h 00h 00h 00h et que la Bank n'est pas sécurisée, il est possible d'utiliser la valeur par défaut ou la valeur réelle.

Le lecteur utilisera les paramètres d'antenne et de puissance associés au port logique en fonction des paramètres par défaut ou définis par les commandes ***Set_RFSettings*** ou ***Set_RFSettings_Saved***. Le paramètre de temps est quant à lui figé.
Si le port logique utilisé dans la commande n'est pas attribué, le lecteur remontera l'erreur **00h 02h**.

Exemple :
00h 08h 00h 05h AAh 55h 00h 15h 01h 09h 04h 00h 11h 22h 33h 44h 55h 66h 77h 88h 00h C3h 00h C2h 00h 00h 00h 00h 00h.

MaskBank: 01h (RFU i.e. Bank EPC).
MaskLen: 9 octets
MaskOffset: 4 octets
Mask: 00h 11h 22h 33h 44h 55h 66h 77h 88h
PWD: 00h 00h 00h 00h
LogicalPort: Port logique 0.
PayLoadMask: C3h

000000	*0 - 0*	*1 - 1*	*0- 0*	*0 - 0*	*1 - 1*
000000	*Kill*	Access	*EPC*	*TID*	*User*

PayLoadAction: C2h

b15 - b10	*b9 - b8*	*1 - 1*	*0 -0*	*0 -0*	*1 - 0*
000000	*Kill*	Access	*EPC*	*TID*	*User*

Bank EPC : Pas de changement

Bank TID : Pas de changement

Bank User : SecuredAccessible

Kill Password : Pas de changement

Access Password : AlwaysNOTAccessible

Read

Description

Commande permettant la lecture de données dans une BankID d'un tag en fonction d'un masque (**Mask**).

Format

Host: 00h 08h 00h 02h AAh 55h L_{out} MaskBank MaskLen MaskOffset Mask BankID BankOffset Len PWD LogicalPort

Lecteur: 00h 02h L_{in} MatchNb $Data_{Len}$ 08h 00h

Paramètres

Code commande : 00h 02h
2 Octets

L_{in}: [01h+**Len**]
2 Octets

L_{out}: Valeur variable en fonction de la longueur du masque. [0Bh+MaskLen] ou [0Ch+MaskLen]
2 Octets

MaskBank: 01h (RFU i.e. Bank EPC).
1 Octet

MaskLen: Longueur du masque EPC en octets [00h ... 1Eh].
1 Octet

MaskOffset: Décalage dans le masque en octets par rapport à la Bank EPC [04h ... 21h].
1 Octet

Mask: Correspond au masque EPC permettant au lecteur de sélectionner le(s) tag(s).
MaskLen Octets

BankID: Octet représentant la Bank devant être lue.
1 Octet

- 00h Bank Reserved
- 01h Bank EPC
- 02h Bank TID
- 03h Bank User Memory

BankOffset: Décalage en mots de 2 octets des données **Data** dans la BankID [00h 00h ... FFh FFh].
1 ou 2 Octets

Len: Longueur des données **Data** exprimées en mots de deux octets devant être lues. [00h ... 20h].
1 Octet

PWD: Mot de 4 octets représentant le mot de passe **Access Password** permettant la lecture. MSB first.
4 Octets

LogicalPort: Octet représentant le port logique sur lequel le lecteur tentera la lecture. [00h ... 0Fh]
1 Octet

MatchNb: Nombre de tags dont l'EPC correspond au masque définit
1 Octet

Data: Valeur correspondante aux données lues dans la Bank.
Len Octets

Remarques

Le masque (**Mask**) n'est effectif uniquement sur la Bank EPC du tag (d'où **MaskBank** = RFU = 01h).

Si l'octet **MatchNb** est supérieur à 01h alors les données **Data** remontées par le lecteur seront celles du dernier tag lu.

Il est important d'indiquer le mot de passe **PWD** (par défaut 00h 00h 00h 00h) même si la Bank n'est pas sécurisée. Dans le cas où celui-ci est différent de 00h 00h 00h 00h et que la Bank n'est pas sécurisée, il est possible d'utiliser la valeur par défaut ou la valeur réelle.

Si aucun tag n'est présent, le code erreur **08h 07h** sera remonté par le lecteur.

Le lecteur utilisera les paramètres d'antenne et de puissance associés au port logique en fonction des paramètres par défaut ou définis par les commandes ***Set_RFSettings*** ou ***Set_RFSettings_Saved***. Le paramètre de temps est quant à lui figé.

Si le port logique utilisé dans la commande n'est pas attribué, le lecteur remontera l'erreur **00h 02h**.

La syntaxe de la commande ***Read*** a changé. Il est possible d'ajouter un octet supplémentaire pour l'information **BankOffset**. Cependant, les anciennes versions de commandes n'incluant qu'un octet pour l'information **BankOffset** seront toujours prises en compte par le lecteur.

Write

Description

Commande permettant l'écriture de données dans une BankID d'un tag en fonction d'un masque (**Mask**).

Format

Host: 00h 08h 00h 03h AAh 55h L_{out} MaskBank MaskLen MaskOffset Mask BankID BankOffset Len Data PWD LogicalPort

Lecteur: 00h 03h 00h 00h 08h 00h

Paramètres

Code commande : 00h 03h
2 Octets

L_{in}: 00h 00h
2 Octets

L_{out}: Valeur variable en fonction de la longueur du masque (**Mask**) et de la valeur à écrire
2 Octets [0Bh + **MaskLen** + (2*__Len__)] ou [0Ch + **MaskLen** + (2*__Len__)].

MaskBank: 01h (RFU i.e. Bank EPC).
1 Octet

MaskLen: Longueur du masque EPC en octets [00h ... 1Eh].
1 Octet

MaskOffset: Décalage dans le masque en octets par rapport à la Bank EPC [04h ... 21h].
1 Octet

Mask: Correspond au masque EPC permettant au lecteur de sélectionner le(s) tag(s).
MaskLen Octets

BankID: Octet représentant la Bank devant être écrite.
1 Octet

- 00h Bank Reserved
- 01h Bank EPC
- 02h Bank TID
- 03h Bank User Memory

BankOffset: Décalage en mots de 2 octets des données **Data** dans la BankID [00h 00h ... FFh FFh].
1 ou 2 Octets

Len: Longueur des données **Data** exprimées en mots de deux octets devant être écrites.
1 Octet [01h…20h]. La valeur maximale étant 32 mots de deux octets.

PWD: Mot de 4 octets représentant le mot de passe **Access Password** permettant l'écriture.
4 Octets MSB first.

LogicalPort: Octet représentant le port logique sur lequel le lecteur tentera l'écriture. [00h ... 0Fh]
1 Octet

Data: Valeur correspondante aux données à écrire dans la Bank.
Len Octets

Remarques

Le masque (**Mask**) n'est effectif uniquement sur la Bank EPC du tag (d'où **MaskBank** = RFU = 01h).

Il est important d'indiquer le mot de passe **PWD** (par défaut 00h 00h 00h 00h) même si la Bank n'est pas sécurisée. Dans le cas où celui-ci est différent de 00h 00h 00h 00h et que la Bank n'est pas sécurisée, il est possible d'utiliser la valeur par défaut ou la valeur réelle.

Le lecteur utilisera les paramètres d'antenne et de puissance associés au port logique en fonction des paramètres par défauts ou définis par les commandes ***Set_RFSettings*** ou ***Set_RFSettings_Saved***. Le paramètre de temps est quant à lui figé.

Si le port logique utilisé dans la commande n'est pas attribué, le lecteur remontera l'erreur **00h 02h**.

La syntaxe de la commande ***Read*** a changé. Il est possible d'ajouter un octet supplémentaire pour l'information **BankOffset**. Cependant, les anciennes versions de commandes n'incluant qu'un octet pour l'information **BankOffset** seront toujours prises en compte par le lecteur.

Exemple :

00h 08h 00h 03h AAh 55h 00h 17h 01h 09h 04h 00h 11h 22h 33h 44h 55h 66h 77h 88h 03h 00h 00h 01h 00h 11h AAh BBh CCh DDh 00h

MaskBank:	01h (RFU i.e. Bank EPC)
MaskLen:	9 octets
MaskOffset:	4 octets
Mask:	00h 11h 22h 33h 44h 55h 66h 77h 88h
BankID:	Bank User Memory
BankOffset:	0 octets
Len:	Figé à 01h.
PWD:	AAh BBh CCh DDh
LogicalPort:	Port logique 0
Data:	00h 11h

Inventory_With_Report

Description

Cette commande retourne la totalité des puces *EPC1 Gen2* détectées par le lecteur lors d'un cycle de lecture.

Format

Host: 00h 08h 00h 11h AAh 55h 00h 04h PARAM1 PARAM2 PARAM3 PARAM4

Lecteur: 00h 11h L_{in} NbTags [EPCLen EPC AntID NbRead RSSI] ^{nbTags} 08h 00h

Paramètres

Code commande : 00h 11h
2 Octets

L_{in}: Valeur variable en fonction de la longueur des codes EPC1 Gen2 ainsi que du nombre total de tags lus. **NbTags + ([EPCLen + EPC + AntID + NbRead +RSSI] x NbTags)**
2 Octets

L_{out}: 00h 00h
2 Octets

NbTags: Cet octet indique le nombre d'identifiants *EPC1 Gen2* différents détectés par le lecteur. [00h … F7h]. 247 tags détectés maximum.
1 Octet

EPCLen : Octet déterminant la longueur (en octets) de l'identifiant *EPC1 Gen2* qui suit. Seuls les identifiants de 96 bits seront détectés.
1 Octet

EPC : Identifiant *EPC1 Gen2* du tag lu. Sa taille dépendant de la valeur **EPCLen**.
EPCLen Octets

AntID : Représente le numéro du premier port logique sur lequel l'identifiant *EPC1 Gen2* a été détecté. Par adéquation, un port logique représente une antenne (voir commandes ***SetRFSettings*** et ***GetRFSettings***)
1 Octet

NbRead : Cet octet représente le nombre de lecture d'un tag *EPC1 Gen2* sur la totalité des ports logiques par commande ***Inventory***.
2 Octets

PARAM1: Cet octet représente l'activation de certaines remontées d'informations.
1 Octet

- 01h RSSI

PARAM2 : 00h. RFU
1 Octet

PARAM3: 00h. RFU
1 Octet

PARAM4: 00h. RFU
1 Octet

RSSI : Cet octet représente l'information RSSI du tag lu. [00h ... FFh]
1 Octet

Remarques

Les paramètres **[EPCLen EPC AntID NbRead RSSI]** sont transmis autant de fois qu'il y a de tags *EPC1 Gen2* lus (**NbTags**). Si aucun tag n'est détecté, ces paramètres ne seront pas transmis. Seul l'octet **NbTags** sera inclus dans la trame avec pour valeur 00h.

L'information *RSSI* (***R**eceived **S**ignal **S**trength **I**ndication*) représente l'amplitude de la réponse du tag lu par le lecteur.
Si un tag est lu plusieurs fois, l'information RSSI remontée sera celle la plus faible détectée au cours des lectures.

V. Codes erreurs

5.1. Codes erreurs lecteurs

Le statut retourné sera donc pour cette famille de la forme 00h Code. Les codes d'erreurs sont indiqués en hexadécimal.

Code	Erreur	Résolution
00h	*OK (Pas d'erreur)*	
02h	*Mauvais paramètre de donnée*	*Mauvais paramètre ou mauvais déchiffrement. Vérifier les paramètres.*
03h	*Erreur de CRC sur la trame*	*Problème de bruit sur la liaison. Mauvaise communication série.*
04h	*Mauvaise longueur de trame reçue*	*Problème de bruit sur la liaison. Mauvaise communication série.*
07h	*Mauvais code de commande*	*Vérifier la commande*
08h	*Mauvais type de commande*	*Vérifier le type de commande*
20h	*Problème hardware du lecteur UHF*	*Problème lié à la qualité ou la connexion d'antenne ou température interne du lecteur (cœur ou amplificateur de puissance) trop élevée.*
D1h	*Problème non récurrent*	
D2h	*Problème non récurrent*	
D3h	*Problème matériel*	*Analyse nécessaire*

5.2. Codes erreurs EPC1 Gen2

Le statut retourné sera donc pour cette famille de la forme 08h Code. Les codes d'erreurs sont indiqués en hexadécimal.

Code	Erreur	Résolution
00h	*OK (Pas d'erreur)*	
01h	*Autre erreur retournée par la puce différente de 03h, 04h et 0Bh.*	*Retenter l'opération.*
02h	*Mauvais paramètre de puce.*	*Vérifier les paramètres de la commande liés à la puce.*
03h	*Adressage mémoire refusé*	*L'adresse mémoire saisie doit être hors de la mémoire disponible sur le tag.*
04h	*Mémoire verrouillée*	*L'accès est refusé car la mémoire est verrouillée. Changer de zone ou déverrouiller la mémoire.*
07h	*Erreur RF*	*Pas de tag détecté ou le masque de détection est trop restrictif. Approcher le tag ou modifier le masque.*
08h	*Erreur RF lors d'une commande **Lock** ou mauvaise valeur de mot de passe pour tag verrouillé*	*Une opération de Lock sur un tag sécurisé (mot de passe différent de 0000) a été tentée avec un mot de passe égal à 0000. Changer ce mot de passe.*
0Bh	*Puissance insuffisante*	*La puissance actuelle du lecteur ne permet pas de finir l'opération en cours. Augmenter la puissance ou rapprocher le tag.*
0Fh	*Mauvais mot de passe*	*Le mot de passe saisit ne permet pas l'accès.*
11h	*Erreur identique 01h lors d'une vérification après commande **Write**.*	*Erreur de vérification*
14h	*Erreur identique 04h lors d'une vérification après commande **Write**.*	*Erreur lors de la vérification de l'opération en cours sur la mémoire verrouillée.*

17h	*Erreur identique 07h lors d'une vérification après commande* ***Write****.*	*Erreur de vérification de l'opération RF en cours. Approcher le tag ou modifier le masque.*
1Bh	*Erreur identique 0Bh lors d'une vérification après commande* ***Write****.*	*Erreur lors de la vérification de l'opération en cours à cause d'une puissance insuffisante.*

VI. Intégration des modules UHF.

Le tableau ci-dessous représente les références des produits *STid* intégrant les modules *URL*, *URM* et *URF*.

URL
URL-W41-A-U04-5AA
URL-W51-A-U04-5AA
URL-Wx1-A-U04-5AA
URL-W41-B-U04-5AA
URL-W51-B-U04-5AA
URL-Wx1-B-U04-5AA
STR-W45-E-U04-5AA
PSION WORKABOUT PRO3

URM
UR1-W42-E-U04-5AA
UR1-W43-E-U04-7AA
UR1-W52-E-U04-5AA
UR1-W53-E-U04-7AA

URF
URC-W42-E-U04-5AA
URC-W43-E-U04-7AA
URC-W52-E-U04-5AA
URC-W53-E-U04-7AA
URD-W42-E-U04-5AA
URD-W43-E-U04-7AA
URD-W44-E-U04-5AA
URD-W45-E-U04-5AA
URD-W48-E-U04-5AA
URD-W52-E-U04-5AA
URD-W53-E-U04-7AA
URD-W54-E-U04-5AA
URD-W55-E-U04-5AA
URD-W58-E-U04-5AA

VII. Révisions

Date	Version	Description
13/08/2009	*1.0*	*Version initiale du document*
02/12/2009	*1.1*	*Modification de la commande **Get_RFSettings***
16/12/2009	*1.2*	*Ajout des commandes **Set_RFSettings_Saved; Set_RFSettings_Defaults***
05/03/2010	*1.3.0*	*Ajout des commandes **Read**, **Write**, **Kill**, **Lock**.* *Modification de la structure document.* *Ajout des codes erreurs.* *Modification des codes commande **Reset_RFSettings**, **Get_RFSettings**, **Set_RFSettings** et **Set_RFSettings_Saved**.* *La commande **Set_RFSettings_Defaults** devient **Reset_RFSettings**.*
25/06/2010	*1.4*	*Modification de la structure document. Précision sur l'utilisation du mot de passe **PWD** pour les Banks non sécurisées sur les commandes **Kill**, **Lock**, **Read** et **Write**.*
19/07/2010	*1.5*	*Précision apportée sur la commande **Inventory** si aucun tag n'est détecté.*
29/07/2010	*1.6*	*Modification de l'octet **RECOM** sur la commande **Kill** (01h devient 00h).*
29/10/2010	*1.7*	*Ajout de la commande **Get_HealthParameters**, des codes erreurs 20h & 21h et implémentation du module URL.* *Ajout du paragraphe § VI*
25/01/2011	*1.8*	*Ajout des commandes **SetOptoOutputParam** et **ChangeRegulation**.* *Modification de la commande **Inventory** sur l'état de l'entrée Input.*
09/03/2011	*1.9*	*Modification de la commande **Autonomous_Output** (b7 de l'octet OutConf sert désormais à activer/désactiver le numéro du port logique remonté)* *Modification de la trame de sortie en mode autonome. Le lecteur peut remonter désormais le numéro de port logique e mode autonome.*
25/03/2011	*1.10*	*Modification des informations PowerAnt et ScanDelay dans la commande **Set_RFSettings**.*
01/07/2011	*1.11*	*Suppression de l'octet Conf dans la commande Autonomous_Output. Remplacé par 00h.*
01/09/2011	*2.0*	*Ajout des commandes **GetInfos**, **SetBaudRate**, **Set485Address** et **Inventory_Report.*** *Modification des syntaxes des commandes **Read** et **Write.*** *La valeur FFh pour l'octet LogicalPort dans les commandes **Read** et **Write permet** d'essayer l'opération d'écriture ou de lecture sur tous les ports actifs.*
15/05/2012	*2.1*	*Ajout de la lecture TID en mode autonomous.* *Modification de la commande **Autonmous_Output**.* *Ajout des commandes **Set_RF_Params** et **Retrieve_RF_Params**.* *Il est possible désormais d'écrire avec la fonction **Write** une donnée de 32 octets.*