

SPECIFICATION DE REFERENCE


PROTOCOLE AFSEC+ ET CARTE INTERFACE COMMUNICATION

DOMAINE D'APPLICATION : Développement

RUBRIQUE : Développement de logiciels pour cible AFSEC+

OBJET : Principe et spécification du protocole entre l'AFSEC+ et la carte Interface Com du E-MICROCOMPT+

G	21/09/23		Ajout messages PACK_IN et PACK_OUT
F	27/04/20		Ajout Gestion Télécommande Alma / Radio fréquence, ajout précision pour D_DATA_TABLE_INDEX
E	22/02/19		Ajout du TAG 'END_OF_RECORD'
D	20/07/17		Ajout menu étendu de saisie sur le prompteur
C	27/03/17		Ajout possibilité de menus dynamiques
B	09/02/17		Ajout exemple DATA_IN
A	08/11/16		Création du document
Ind.	Date	Réf.	Nature des modifications
Rédacteur : DD/DRA		Vérificateur : DRA/DD	Approbateur : ZZ
		Organisme :	
DD 21/09/2023			

	SR DEV 0006 G du 21.09.2023	Page 1/52
	Protocole AFSEC+/Télécommande Alma vs iCom Service Développement	

SOMMAIRE

1	Référentiels et documents associés	6
2	Objet.....	6
2.1	Carte Interface Com et AFSEC+.....	6
2.2	Télécommande Alma.....	6
3	Terminologie.....	6
4	L'interfaces physiques	7
4.1	ICom vs l'AFSEC+	7
4.2	ICom vs Télécommande.....	7
5	Solution technique.....	8
5.1	Description du système	8
5.2	Description du protocole	9
5.2.1	Encapsulation des protocoles	9
5.2.2	Cinématique des échanges.....	10
5.2.3	Contraintes temporelles pour une communication UART	12
5.2.4	Contraintes temporelles pour une communication radio fréquence	13
5.2.5	Transaction, conversation et contexte.....	14
5.2.6	Schéma TLV	14
5.3	Règles de gestion pour les conversations client / ICom	15
6	Messagerie	16
6.1	Message ALIVE (signe de vie)	20
6.1.1	Message AF_ALIVE ou TL_ALIVE (0x00).....	20
6.1.2	Message IC_ALIVE (0x80)	20
6.2	Message INIT.....	20
6.2.1	Message AF_INIT ou TL_INIT (0x01)	21
6.2.2	Message IC_INIT (0x81)	21
6.3	Message MENU (MC+ Uniquement).....	22
6.3.1	Usage des menus.....	22
6.3.2	Message AF_MENU (0x02).....	23
6.3.3	Message IC_MENU (0x82)	24
6.3.4	Exemples de conversations	25
6.4	RF_MENU (RF uniquement)	26
6.4.1	Message TL_RFMENU (0x0A)	26
6.5	Message DATA	27
6.5.1	Message AF_DATA_OUT ou TL_DATA_OUT (0x03).....	28
6.5.2	Message IC_DATA_OUT (0x83)	29
6.5.3	Message IC_DATA_IN (0x84)	29

6.5.4	Message AF_DATA_IN (0x04)	30
6.5.5	Cinématique des conversations DATA_IN et DATA_OUT	30
6.5.5.1	Règles de gestion	30
6.5.5.2	Exemples de conversations	30
6.5.6	Message AF_DATA_OUT_TABLE_INDEX (0x05) (MC+ uniquement)	32
6.5.7	Message IC_DATA_OUT_TABLE_INDEX (0x85) (MC+ uniquement)	32
6.6	Message DATA_IN_REQ (RF Uniquement)	32
6.6.1	Message TL_DATA_IN_REQ (0x08)	33
6.6.2	Message IC_DATA_IN_REQ (0x88)	33
6.6.3	Message IC_DATA_IN_RES (0x89)	34
6.7	Messages PACK (0x0B, 0x0C, 0x8B et 0x8C)	34
6.7.1	Message AF_PACK_OUT (0x0B) et IC_PACK_OUT (0x8B)	36
6.7.2	Message IC_PACK_IN (0x8C) et AF_PACK_IN (0x0C)	36
6.7.3	Exemples de conversations	36
6.8	Message DOWNLOAD (MC+ uniquement)	37
6.8.1	Message IC_DOWNLOAD (0x86)	37
6.8.2	AF_DOWNLOAD (0x06)	38
6.9	Message SYNC (RF Uniquement)	38
6.9.1	Message TL_SYNC (0x07)	39
6.9.2	Message IC_SYNC (0x87)	39
6.10	Message TEST	39
6.10.1	Message AF_TEST (0x7F)	39
6.10.2	Message IC_TEST (0xFF)	40
6.10.3	Cinématique des conversations de TEST	40
7	ANNEXES	41
7.1	Codage de la longueur et format d'une donnée TLV	41
7.2	Tags des données internes (D_xxxx)	41
7.3	Table des bits pour les pictogrammes (D_MENU_PICTOS)	44
7.4	Numérotation des « zones » pour les données applicatives	45
7.5	Table des codes d'erreur lors d'une transmission de donnée applicative (D_DATA_ERROR)	46
7.6	Structure d'un index pour les données (D_DATA_TABLE_INDEX)	46
7.7	Masque de saisie D_MENU_INPUT_MASK pour IC_MENU	47
7.8	Index d'affichage DISP_INDEX	47
7.9	Traces exemples d'échanges AFSEC+/iCom	47
7.9.1	Conversation DATA_OUT	47
7.9.2	Conversation DATA_IN	48
7.9.3	Conversation MENU	48
7.9.4	Conversation DOWNLOAD	51

7.10 Traces exemples d'échanges ICom/Télécommande 52

7.10.1 Conversation SYNC 52

7.10.2 Conversation DATA_OUT 52

7.10.3 Conversation DATA_IN 52

7.10.4 Conversation DATA_IN_REQ 52

1 REFERENTIELS ET DOCUMENTS ASSOCIES

- ⇒ **NT 981/01** : Description de l'architecture matériel et logiciel du E-Microcompt+
- ⇒ **NT 9547** : Description de l'architecture logiciel de la Télécommande Alma
- ⇒ **SR DEV 0004** : Protocole de communication entre l'outil µConfig et un équipement
- ⇒ **SR DEV 0008** : ICom pour les développeurs (guide pour le portage d'applications)
- ⇒ **SL 4000** : Répertoire DOCS de l'archivage du logiciel #4000 = Résident AFSEC+

2 OBJET

2.1 Carte Interface Com et AFSEC+

La « Carte Interface Com » (ICom) est une extension de l'AFSEC+ qui permet d'échanger des données avec le monde extérieur en utilisant des moyens de communication modernes (Ethernet, Wifi, Bluetooth, RFID NFC, GSM et LoRa).

Une interface matérielle intégrant des E/S Tout ou Rien et une UART est ajoutée à la partie existante de la solution AFSEC+ pour échanger avec la « Carte Interface Com ».

Le logiciel « résident » et les applications développées pour l'AFSEC+ doivent donc évoluer pour intégrer et prendre en compte ces nouveaux canaux de communication.

Ce document décrit les principes et le protocole mis en place au niveau du résident et des applications AFSEC+ pour ces nouvelles fonctionnalités.

Ce document ne décrit pas la solution matérielle ou le logiciel en place dans la « Carte Interface Com ».

2.2 Télécommande Alma


La télécommande Alma est un appareil autonome pouvant contrôler plusieurs fonctionnalités d'une application MC+ à distance.

Cette télécommande communique par liaison sans fils suivant la technologie LoRa avec l'ICom. Les informations qui transitent dans ce canal RF sont encodées suivant le protocole de ce document et sécurisées par chiffrement AES-256.

Charge ensuite à l'ICom de router les informations échangées entre la télécommande et l'application de l'AFSEC+.

3 TERMINOLOGIE

- ⇒ **ICom** : Abréviation pour la « Carte Interface Com »
- ⇒ **Résident** : Le « Résident » est la partie logicielle « de base » de la carte AFSEC+. Il est systématiquement présent dans toutes les solutions AFSEC+. C'est ce logiciel qui est démarré à la mise sous tension de l'AFSEC+

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 6/52
	Service Développement	

- ⇒ **Application** ou **Applicatif** AFSEC+ : C'est un logiciel AFSEC+ dédié pour une solution client. Il est démarré par le résident en mode 'application' (non 'résident'). Il lui est possible d'accéder aux services proposés par le Résident
- ⇒ **Mode Résident** : Le mode « résident » est consécutif à une position précise des switches de l'AFSEC+. Dans ce mode, le résident « garde la main » et propose des actions particulières à l'utilisateur via l'outil 'TéléConfig' sur une liaison série : téléchargement de logiciels, de lots de tickets, traduction ou tests de la carte, etc. Les menus (boutons poussoirs et affichage correspondant) ne sont pas actifs dans ce mode. Lorsque le résident n'est pas en mode résident, il démarre l'application.
- ⇒ **Mode Métrologique, Mode Superviseur** ou **Mode Utilisateur** : Ce sont 3 modes de fonctionnement qui sont pris en charge par l'Applicatif. Ils correspondent également à des positions particulières des switches de l'AFSEC+ :
 - Mode Métrologique : Déverrouillage de l'E2PROM et possibilité de modification des paramètres métrologiques de l'application en E2PROM
 - Mode Superviseur : Possibilité de modification des paramètres en RAM secourues
 - Mode Utilisateur : Fonctionnement « normal » de l'application pour l'utilisateur final
- ⇒ **TéléConfig** (protocole et outil) : Solution qui permet de communiquer via une liaison série avec le résident de l'AFSEC+ (voir les archives des logiciels SDK+). En particulier pour le téléchargement de programmes (résident ou applications), de catalogue de traduction ou de lots de tickets
- ⇒ **µConfig** (protocole et outil) : Solution qui permet d'identifier et d'échanger des blocs de « données unitaires » entre l'application (tous les modes) et un intervenant externe. C'est le catalogue qui référence toutes les « données unitaires » de toutes les applications (SR DEV 0004) qui est utilisé ici
- ⇒ **Donnée unitaire** : C'est une information atomique qui a un sens pour le résident, l'application ou les utilisateurs
- ⇒ **RF** : Radio fréquence
- ⇒ **LoRa** : Technologie de la radio fréquence

4 L'INTERFACES PHYSIQUES

4.1 ICom vs l'AFSEC+


C'est la liaison série identifiée par « COM3 » de l'AFSEC+ qui est utilisée ici.

Configuration en RS232C : 115 200 Bd, 8 bits data, 1 start, 1 stop, pas de parité, pas de contrôle de flux matériel ou logiciel

4.2 ICom vs Télécommande

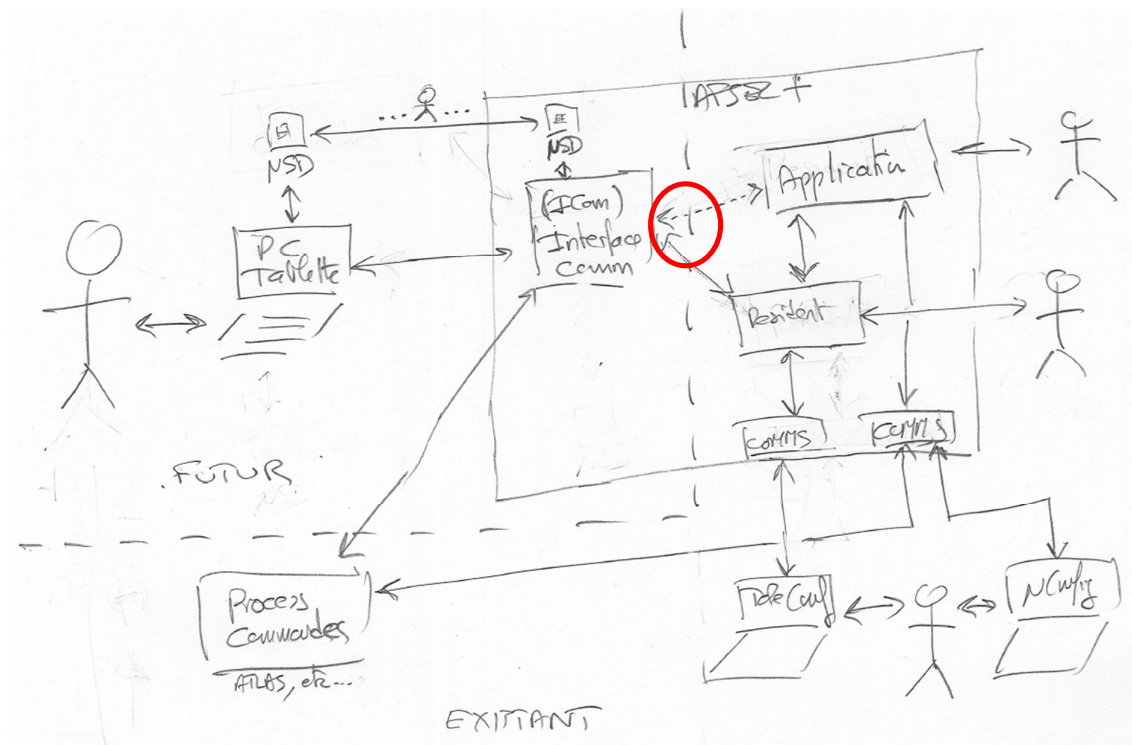
C'est une liaison RF suivant les configurations ci-dessous :

Config : Freq : 868MHz ; SF : 5 ; BP :500kHz ; CR : 4/8

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 7/52
	Service Développement	

5 SOLUTION TECHNIQUE

5.1 Description du système



Le protocole décrit ici est celui (en rouge) entre la partie Résident (périmètre 'existant') et la partie ICom (périmètre 'futur').

La révision F de ce document étend l'utilisation du protocole à une télécommande ALMA en interface avec l'ICom par une communication radio fréquence LoRa.

Nota : Le schéma fait apparaître une communication directe (en pointillé) entre l'application et l'ICom mais de fait, cette communication s'effectue toujours sous le contrôle du résident (via les primitives du SDK pour l'application – Voir le détail dans l'analyse fonctionnelle du résident).

5.2 Description du protocole

Le protocole décrit ici correspond à la **version 0.0.3**.

Les paragraphes avec la mention « *MC+ uniquement* » sont uniquement utilisées entre l'ICom et l'AFSEC+ »

Les paragraphes avec la mention « *RF uniquement* » sont uniquement utilisées entre l'ICom et la Télécommande

5.2.1 Encapsulation des protocoles

Il y a 3 niveaux d'encapsulation décrits dans ce document :

- Des messages de la forme **Type-Longueur-Valeur** qui sont des données d'application.
La sémantique de ces messages est décrite au § 'Schéma TLV' ci-dessous

- Selon le support physique utilisé, ces messages TLV sont encapsulés différemment :

- Pour une communication série (UART), on encapsule ensuite par :

STX [message TLV] XOR ETX

Voir la description au § Cinématique des échanges ci-dessous

- Pour une communication par radio fréquence (RF), les étapes suivantes ont lieu :

- Un compteur (1 octet) incrémenté à chaque envoi de nouveau message est ajouté en tête du bloc précédent comme illustré ci-dessous :

Count STX [message TLV] XOR ETX

- Une opération de chiffrement AES 256 est ensuite effectuée sur ce bloc en utilisant la clef de chiffrement D_CIPHER_KEY échangée lors de la conversion SYNC.

Ceci nécessite que le bloc est une longueur multiple de 16 octets.


Il faut donc combler cette trame par des caractères nuls (0x00) jusqu'à obtenir une longueur totale multiple de 16 octets.

Count STX [message TLV] XOR ETX [0 0 0 ... 0]

Il y aura alors à la réception une étape de suppression des zéros de queue après le décryptage et avant le décodage TLV.

NOTE : s'il s'agit d'un message TL_SYNC, alors aller directement à l'étape ci-dessous (pas de chiffrement).

S'il n'y a pas eu de conversation SYNC pour échanger la clef de chiffrement alors ignorer le traitement et sortir.

	SR DEV 0006 G du 21.09.2023	Page 9/52
	Protocole AFSEC+/Télécommande Alma vs iCom	
	Service Développement	

- Un identifiant (2 octets) est ajouté en tête du bloc chiffré (voir le détail dans le § Cinématique des échanges ci-après) :

```
TypeID Encrypted{{Count STX [message TLV] XOR ETX [0
0 0 ... 0]}}
```

TypeID contient le numéro de série de l'appareil qui émet le message. Hormis le message TL_SYNC, tous les messages en provenance d'un appareil non identifié sont ignorés (voir le détail dans le § Message SYNC ci-dessous).

- Enfin, la couche LoRa reprend cet ensemble (payload) pour l'encapsuler à nouveau par :

```
Préambule Header [Payload] CRC
```

Voir la description au § Cinématique des échanges ci-dessous

Le décodage des trames est réalisé en effectuant les mêmes opérations en ordre inverse.

Remarque : A terme, il conviendrait peut-être de faire 3 documents distincts pour clarifier la lecture.

5.2.2 Cinématique des échanges

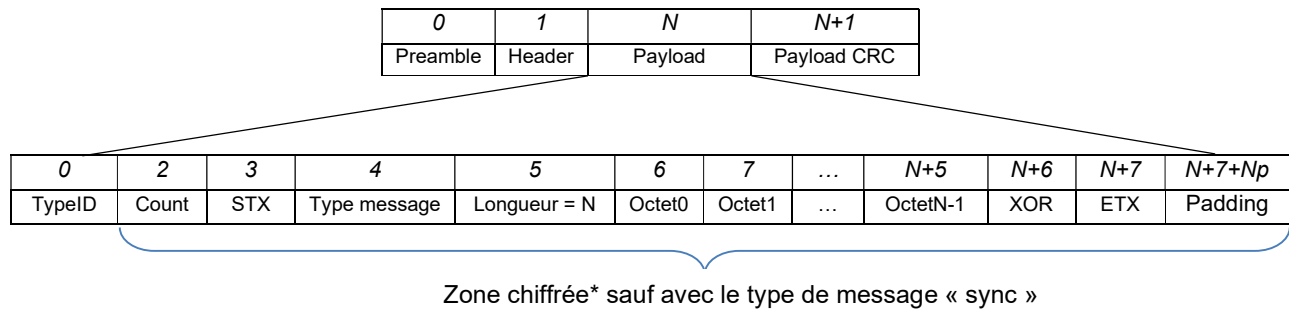
Tous les échanges sont réalisés à l'initiative de la partie AFSEC+ ou de la télécommande ('clients' possibles de l'ICom).

Important : Pour une communication Radio Fréquence avec une télécommande, tant que les numéros de séries des équipements télécommande et ICom ainsi que la clef de chiffrement ne sont pas clairement échangés par une conversion SYNC (voir le détail dans le § Message SYNC ci-dessous) et mémorisé par les appareils, aucun autre échange ne peut se faire car tous les autres messages sont alors ignorés.

L'AFSEC+ envoie sur l'UART un message de la forme :

0	1	2	3	4	...	N + 2	N + 3	N + 4
STX	Type message	Longueur = N	Octet0	Octet1	...	OctetN-1	XOR	ETX

En mode RF, l'ICom ou la télécommande envoie un message de la forme :



* : Attention cette zone est toujours un multiple de 16 octets. (nécessaire pour le chiffrement AES256)

Où :

- Preamble : Octets gérée par la couche physique du protocole LoRa
 - Header : Entête gérée par la couche physique du protocole LoRa
 - TypeID (2 octets) : Type (1bit) et numéro de série du périphérique (15 bits). Le type est sur le bit de poids fort
 0_b = Maître/Télécommande ; 1_b = Esclave/ICom
 Si le TypeID ne correspond pas à l'appareil enregistré au niveau de l'ICOM, le reste du message n'est pas traité
 Si le TypeID n'est pas initialisé dans l'ICOM seul un message TL_SYNC est pris en compte pour définir ce TypeID
 - Count (octet) : c'est un compteur incrémental sur à chaque envoi de paquet (1). (Cet octet permet de vérifier s'il n'y a pas eu de données perdues entre deux paquets.)
 - Payload CRC (2 octets) : Octets gérés par la couche physique du protocole LoRa
 - STX (octet) = Code binaire 0x02 (mode UART uniquement)
 - Type message (octet) = Indication du type de message entre 0 et 127 (voir signification ci-dessous). Les données transmises (Data0... DataN) dépendent de ce type. Un même type de message avec le bit 7 (bit de poids fort) à 1 est utilisé pour une réponse de l'ICom à ce message
 - Longueur (octet) = Entre 0 et 250 qui indique la longueur des données à suivre. Cette longueur peut être 0 (pas de donnée à suivre). La longueur maximum est de 250 octets de données
 - Octet0 à OctetN-1 = N octets de données du message où N est la longueur spécifiée.
- En mode RF : Pour avoir une longueur de trame complète en un multiple de 16 octets cette zone peut être complétée avec des caractères NUL.
- Au décodage, on stoppe dès le premier tag NUL trouvé en s'assurant que le reste des valeurs ne contient bien que des NUL.

- XOR (octet) : c'est une opération XOR des tous les octets de la trame TLV depuis le "type message" jusqu'au dernier octet de données inclus (le STX en position 0 est exclu ainsi bien sûr que l'ETX final).
- ETX (octet) = Code binaire 0x03

Nota : La longueur max des données $N = 250$, donc la longueur max du message est $250 + 5 = 255$ octets en mode UART

Les réponses possibles de l'ICom sont :

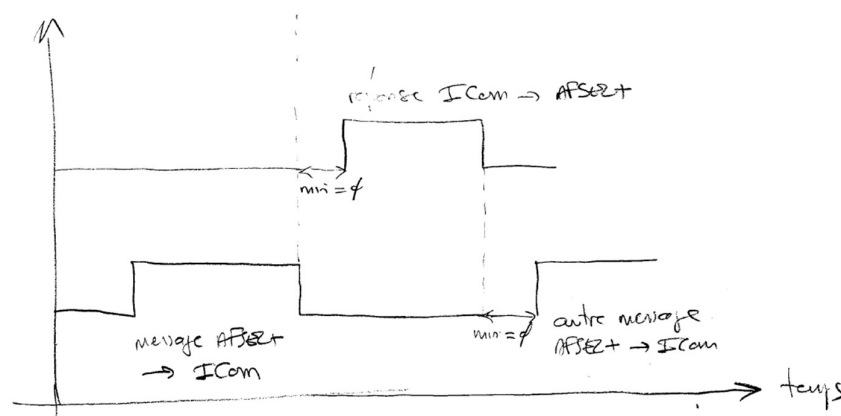
- NAK (octet seul) = Code binaire 0x15. Indique que le message n'a pas été reçu correctement et donc non traité. L'AFSEC+ doit alors réémettre le message pour qu'il soit traité
- ACK (octet seul) = Code binaire 0x06. Indique que le message a été reçu et correctement traitement traité. Il s'agit d'une réponse courte de l'ICom
- Un message de même structure que celui émis par l'AFSEC+ dans lequel le bit 7 (bit de poids fort). Il s'agit d'une réponse longue de l'ICom
- Le cas échéant, un message de même structure émis par l'ICom. Il s'agit ici d'une réponse adaptée de l'ICom ; Par exemple, une invite de l'AFSEC+ à l'ICom transmettre des données signalées prêtes dans l'interface

C'est la nature de la conversation engagée qui détermine si la réponse de l'ICom peut être courte (ACK seulement) car pas de donnée spécifique attendue en retour par l'AFSEC+ ou longue car des données sont attendues en réponse à la requête.

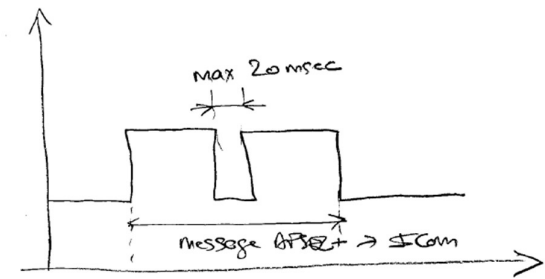
La réponse de l'ICom à un envoi de l'AFSEC+ ou de la télécommande peut se réaliser sans délai (pas de temporisation min. à respecter pour la réponse).

5.2.3 Contraintes temporelles pour une communication UART

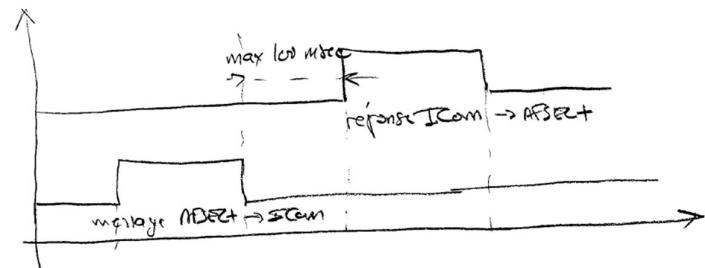
Un nouvel envoi de l'AFSEC+ après la réponse de l'ICom peut se réaliser sans délai (dans la pratique, ce ne sera pas le cas, au moins 20 msec...)



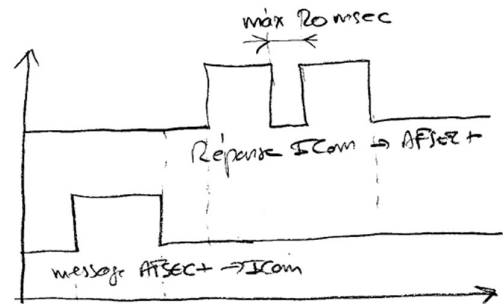
Un message émis par l'AFSEC+ avec un silence de plus de 20 msec inter caractères ou un XOR incorrect peut être considéré comme non traitable par l'ICom qui doit alors envoyer NAK.



Un message émis par l'AFSEC+ qui n'a pas de réponse de l'ICom après un timeout de 100 msec est considéré comme perdu. Il doit être réémis par l'AFSEC+.



Une réponse longue faite par l'ICom avec un silence de plus de 20 msec inter caractères ou un XOR incorrect peut être considérée comme non correcte par l'AFSEC+ qui doit alors renvoyer le message.



5.2.4 Contraintes temporelles pour une communication radio fréquence

L'utilisation de la bande passante est limitée avec l'usage du LoRa.

C'est à l'initiateur des échanges, ci la télécommande ALMA, de réguler le flux des communications avec l'ICom pour satisfaire à cette contrainte.

5.2.5 Transaction, conversation et contexte

Une **transaction** client/ICom se fait toujours par la bonne réalisation de :

- Envoi correct d'un message client -> ICom
- Réponse correcte par un message ICom -> client

Ici, client étant l'AFSEC+ ou la télécommande.

Certains échanges de données nécessitent plusieurs transactions. Un transfert de fichier, par exemple. Dans ce cas, on parle d'une **conversation** et les données échangées permettent de suivre le fil de cette conversation.

Lors d'une conversation, un **contexte** est établi au fil des différentes conversations. Ce contexte est valable pendant toute la conversation et il est RAZé au début d'une nouvelle conversation.

Typiquement, une conversation engagée pour transmettre des données applicatives peut se dérouler sur plusieurs transactions. La 'zone' en cours pour les données applicatives transmises n'est pas signifiée à chaque transaction. Il s'agit d'une information 'contextuelle'.

5.2.6 Schéma TLV

Les messages échangés entre l'AFSEC+ et l'ICom sont 'tagués' selon le schéma TLV (Tag/Length/Value) :

T	L = N	V0	V1	V2	...	VN-1
Tag (1 octet)	Longueur & Format (1 octet qui détermine la valeur 'N' du nombre d'octets à suivre)	Octet0 de la valeur	Octet1 de la valeur	Octet2 de la valeur	...	OctetN-1 de la valeur

Où :


- **T** est le Tag codé sur un octet selon la table donnée en annexe
- **L** est la longueur et le format de la donnée à suivre codée sur un selon la table donnée en annexe
- **V** est la valeur (Value) de la donnée codée sur autant d'octets que spécifiée dans L. Ce codage est binaire (big-endian)

Chaque information Data0, Data1, ... DataN contenue dans le message sont au format TLV.

Elles sont placées continument dans le corps du message pour remplir les N octets des données utiles :

Le protocole permet ainsi d'avoir plusieurs informations codées TLV dans la zone de données d'un message dans la limite des 250 octets max admissibles :

Par exemple, message avec 3 informations distinctes :

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 14/52
	Service Développement	

Octet 0	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7	Octet 8	Octet 9	Octet 10
T1	L1	V1	T2	L2	V2		T3	L3	V3	
T1	1	V1 octet0	T2	2	V2 octet0	V2 octet1	T3	2	V3 octet0	V3 octet1

//\ Attention /\ : L'ordre dans lequel les informations sont placées est important. Voir les conversations DATA_IN et DATA_OUT.

Techniquement, il n'est pas possible d'avoir plus de 83 TLV dans un même message (cas où ils seraient tous d'une longueur nulle).

Ce schéma permet également qu'une information puisse être ajoutée ou supprimée d'un message ou même qu'un message soit traité sans que le destinataire connaisse toutes les informations transmises (compatibilité ascendante non stricte, pas besoin de mise à jour de firmwares).

5.3 Règles de gestion pour les conversations client / ICom

Pour plus de clarté, les noms symboliques des messages sont préfixés par l'appareil à l'origine du message :

- AF_xxx : Message xxx envoyé par la partie AFSEC+ (application) du MC+ à l'ICom
- TL_xxx : Message xxx envoyé par la télécommande ALMA à l'ICom
- IC_xxx : Réponse de l'ICom au message AF_xxx ou TL_xxx

Certains messages sont communs entre l'AFSEC+ et la télécommande. Par exemple, les messages TC_ALIVE et AF_ALIVE ont le même contenu et portent donc le même numéro.

Les règles de gestion pour les conversations entre un client (AFSEC+ ou télécommande) et l'ICom sont :

- Le client débute toujours par un message AF_INIT ou TL_INIT et s'attend à un message IC_INIT au démarrage de la communication ou lorsqu'un problème est détecté sur le fil de la conversation
- Le client initie une conversation par un message AF_xxx (AFSEC+) ou TL_xxx (télécommande) et s'attend à des réponses de l'ICom dans le scope de la conversation telle que décrit dans la messagerie. Un message AF_INIT, TL_INIT, TL_ALIVE ou AF_ALIVE met fin à cette conversation
- Un message AF_ALIVE envoyé par l'AFSEC+ ou un message TL_ALIVE envoyé par la télécommande est une invite à parler pour l'ICom qui peut alors engager une conversation. L'AFSEC+ entretient cette conversation tant qu'elle demeure

dans le scope de la conversation telle que décrit dans la messagerie. Un message AF_INIT, TL_INIT, TL_ALIVE ou AF_ALIVE met fin à cette conversation


- Si l'AFSEC+ ou la télécommande détecte un message inattendu ou erroné lors d'une conversation, un message AF_INIT ou TL_INIT est émis pour redémarrer l'ensemble et mettre fin à toute conversation qui serait en cours
- Lorsqu'une conversation est engagée par l'AFSEC+ ou la télécommande, elle est nécessairement terminée par un message AF_ALIVE ou TL_ALIVE et le client ne peut pas engager une autre conversation dans la foulée
- De même, lorsqu'une conversation est engagée par l'ICom, elle est nécessairement terminée par un message de l'ICom dans le scope de la conversation pour y mettre fin. L'AFSEC+ ou la télécommande décide alors de la relance d'une conversation à suivre ; Soit un message AF_xxx ou TL_xxx pour une conversation ; Soit un message AF_ALIVE ou TL_ALIVE pour inviter l'ICom à entamer une nouvelle conversation. Dans tous les cas, si l'ICom ne peut pas enchaîner 2 conversations consécutives à son initiative sans passer par une invite AF_ALIVE ou TL_ALIVE (ce serait rejeté par un INIT)
- L'AFSEC+ et la télécommande comme l'ICom s'engagent à ne pas entretenir des conversations trop longues ni sans fin afin de ne pas bloquer les échanges pour d'autres conversations (aucun contrôle n'est réalisé sur ce point).

Des exceptions à cette règle pour les conversations MENU et DOWNLOAD qui sont préemptives.

- En mode RF, l'ICom et la télécommande s'engagent à entretenir des conversions qui n'entament pas la bande passante au-delà de la limite tolérée par le LoRa.

6 MESSAGERIE

Symbole	Valeur (AF → IC ou TL → IC)	Valeur (IC → AF ou IC → TL)	Description
ALIVE	AF_ALIVE TL_ALIVE 0x00	IC_ALIVE 0x80	Signe de vie
INIT	AF_INIT TL_INIT 0x01	IC_INIT 0x81	Donnée d'initialisation
MENU	AF_MENU 0x02	IC_MENU 0x82	Gestion des menus entre l'AFSEC+ et l'ICom (<i>MC+</i> <i>uniquement</i>)
DATA_OUT	AF_DATA_OUT TL_DATA_OUT 0x03	IC_DATA_OUT 0x83	Données générées par l'AFSEC+ ou de la Télécommande pour l'ICom

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 16/52
	Service Développement	

Symbole	Valeur (AF → IC ou TL → IC)	Valeur (IC → AF ou IC → TL)	Description
DATA_IN	AF_DATA_IN 0x04	IC_DATA_IN 0x84	Données générées par ICom pour l'AFSEC+ ou la Télécommande
<p>The diagram shows two scenarios of data exchange between RCT5, iCom, and AFSEC+.</p> <p>Scenario 1 (Top):</p> <ul style="list-style-type: none"> RCT5 sends TL_DATA_OUT 0x03 to iCom. iCom sends IC_DATA_OUT 0x83 or ACK back to RCT5. iCom sends AF_ALIVE to AFSEC+. AFSEC+ sends IC_DATA_IN 0x84 to iCom. AFSEC+ sends IC_DATA_IN 0x04, ACK or NAK back to iCom. <p>Scenario 2 (Bottom):</p> <ul style="list-style-type: none"> AFSEC+ sends IC_DATA_OUT 0x03 to iCom. iCom sends IC_DATA_OUT 0x83, ACK or NAK back to AFSEC+. 			
DATA_OUT_TABLE_INDEX	AF_DATA_OUT_TABLE_INDEX 0x05	IC_DATA_OUT_TABLE_INDEX 0x85	Ce message est émis par l'AFSEC+ pour obtenir de l'ICom le premier et le dernier index connus pour une zone organisée en table. (MC+ uniquement)
DOWNLOAD	AF_DOWNLOAD 0x06	IC_DOWNLOAD 0x86	Téléchargement d'une mise à jour de l'ICom vers l'AFSEC+ ou de l'ICom vers la télécommande
SYNC	TL_SYNC 0x07	IC_SYNC 0x87	Appairage entre une télécommande et l'ICom (RF uniquement)

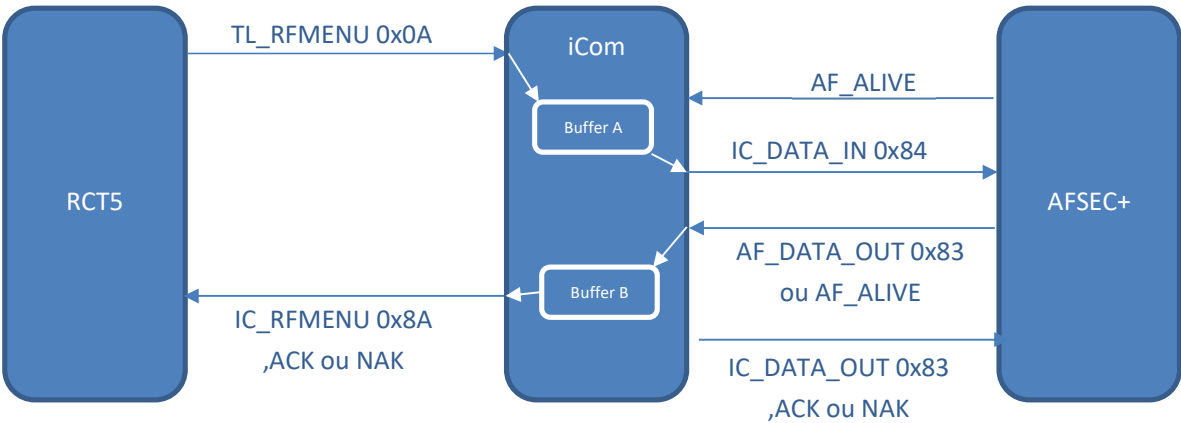
Symbole	Valeur (AF → IC ou TL → IC)	Valeur (IC → AF ou IC → TL)	Description
DATA_IN_REQ	TL_DATA_IN_REQ 0x08	IC_DATA_IN_REQ 0x88	Requête de la télécommande pour recevoir des DATA_IN de façon cyclique ou juste une seul fois (RF uniquement)
DATA_IN_RES		IC_DATA_IN_RES 0x89	Réponses à une requête Provenant de TL_DATA_IN_REQ
<p>The diagram illustrates the communication flow between three main components: RCT5, iCom, and AFSEC+. RCT5 is on the left, iCom is in the center, and AFSEC+ is on the right. iCom contains two sub-components: Slot_x and Database. The communication is as follows: RCT5 sends a TL_ALIVE signal to iCom. iCom sends an IC_DATA_IN_RES 0x89 signal back to RCT5. AFSEC+ sends an AF_DATA_OUT 0x03 signal to the Database in iCom. The Database in iCom then sends an IC_DATA_OUT 0x83 signal to AFSEC+, which includes an ACK or NAK response.</p>			
RFMENU	TL_RFMENU 0x0A	IC_RFMENU 0x8A	Gestion des menus entre l'iCom et la télécommande (RF Uniquement) Pour ce type de message l'iCom fait uniquement une redirection des données avec un repackaging de la trame

Notes :

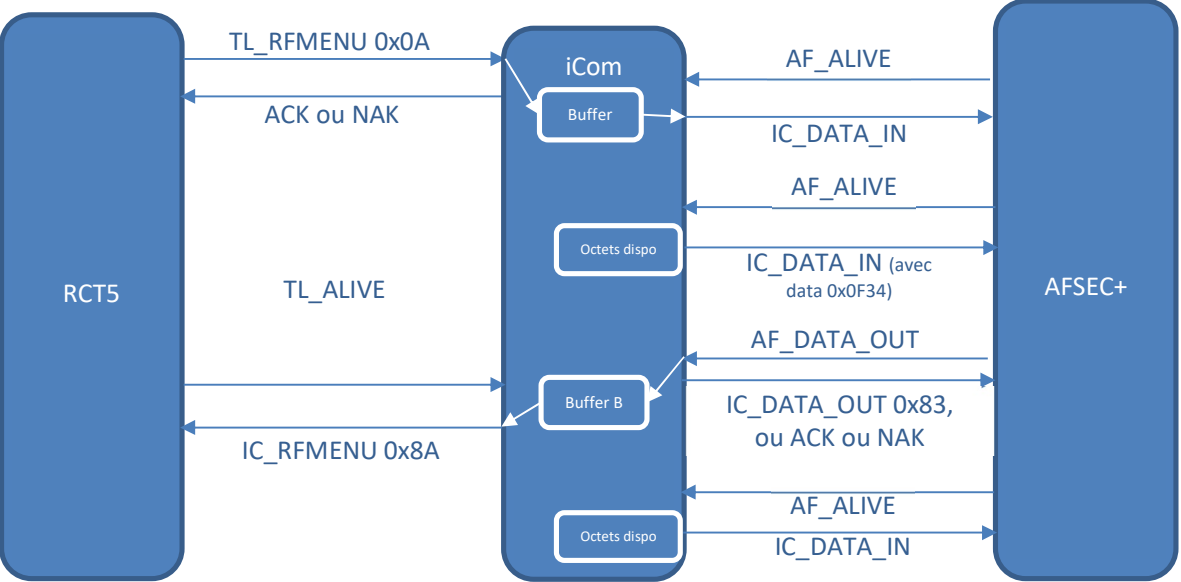
Mise à part l'entête les données dans les trames TL_RFMENU et IC_DATA_IN sont identiques.
Mise à part l'entête les données dans les trames IC_DATA_OUT et IC_RFMENU sont identiques.
Dans ce type de message le rôle de l'iCom va être de rediriger les trames avec les ID suivants : (voir Database ICom)
Pour lancer la synchronisation des libellés, la RCT5 va envoyer la donnée 0x0F32 (CRC des libellés en RAM) à l'AFSEC. Si la donnée reçue est identique à celle qui est stockée dans sa mémoire, celle-ci ne fait rien. Sinon la procédure de synchronisation se lance.
Lorsque les données 0x0F32 et 0x0F33 sont remontées vers la RCT5, l'iCom qui va informer l'AFSEC du nombre d'octet disponible dans le Buffer B (0x0F34). C'est à l'AFSEC d'adapter le débit montant par rapport à la disponibilité du buffer B.
La remontée des données 0x0F32 et 0x0F33 peut-être à l'initiative de l'AFSEC+

- 0x0F30(Affichage des menus) (AFSEC+ → RCT5)
- 0x3F30 (Mapping BP) (RCT5 → AFSEC+)
- 0x0F3A (Retour choix utilisateur) (RCT5 → AFSEC+)
- 0x0F31 (CRC des libellés en RAM) (RCT5 → AFSEC+)
- 0x0F32 (Index des libelles en RAM) (AFSEC+ → RCT5)
- 0x0F33 (Chaine de caractère correspondant à l'index en RAM) (AFSEC+ → RCT5)
- 0x0F34 (Nombre d'octet dispo dans le buffer de transmission « Buffer B ») (ICOM → AFSEC+)

Mode échange menus AFSEC+ vs RCT5



Mode échange libellés AFSEC+ vs RCT5



Symbole	Valeur (AF → IC ou TL → IC)	Valeur (IC → AF ou IC → TL)	Description
PACK_OUT	AF_PACK_OUT 0x0B	IC_PACK_OUT 0x8B	Portion d'un 'data-pack' généré par l'AFSEC+ pour l'ICOM
PACK_IN	AD_PACK_IN 0x0C	IC_PACK_IN 0x8C	Portion d'un 'data-pack' généré par l'ICOM pour l'AFSEC+
TEST	0x7F	0xFF	Trame de test

6.1 Message ALIVE (signe de vie)

Ce message permet à l'AFSEC+ ou Télécommande de :

- Entretenir la conversation avec l'ICOM : Ce message est envoyé périodiquement lorsqu'aucune autre conversation n'est en cours ou à engager
- De vérifier la présence de l'ICOM
- De proposer à l'ICOM d'engager une conversation à sa convenance

6.1.1 Message AF_ALIVE ou TL_ALIVE (0x00)

Message contenant généralement aucune donnée (longueur des données à suivre = 0).

Lorsque le mode de l'AFSEC+ change, principalement les changements de mode de l'application entre les modes métrologique/superviseur/utilisateur, ce message intègre également l'information D_MODE_AFSEC

6.1.2 Message IC_ALIVE (0x80)


Plusieurs réponses possibles de l'ICOM à ce message :

- ACK : Interprété comme un signe de vie de l'ICOM
- Message de données internes (TLV) avec
IC_ALIVE [0x02 0x80 0x00 0x80 0x03] : Également interprété
comme un signe de vie de l'ICOM
- Autres messages IC_XXX autorisés par le protocole afin que l'ICOM entame
une conversation qu'elle souhaite.
Voir le détail ci-après dans chaque paragraphe relatif aux conversations.

6.2 Message INIT

Ce message est envoyé par l'AFSEC+ ou la télécommande après un démarrage ou une demande de RESET de l'ICOM :

- AF_INIT : Message de l'AFSEC+ -> ICOM

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 20/52
	Service Développement	

- IC_INIT : Message réponse de l'ICom -> AFSEC+

Les versions supportées du protocole sont échangées. Si la version est différente, la communication entre l'AFSEC+ et l'ICom est impossible (à confirmer). Chaque partie signale une erreur et il n'y a pas d'autres échanges.

6.2.1 Message AF_INIT ou TL_INIT (0x01)

Message avec les données possibles :

- D_PROTOCOL_VERSION : Version x.y.z du protocole AFSEC+/ICom utilisée par l'AFSEC+
- D_RESIDENT_VERSION : Version du résident de l'AFSEC+
- D_APPLI_NUMBER (*) : Numéro logiciel application de l'AFSEC+ ou Télécommande
- D_APPLI_VERSION (*) : Numéro de version du logiciel application de l'AFSEC+ ou Télécommande
- D_APPLI_CONFIG (*) : Identifiant de la configuration 'usine' de l'application AFSEC+
- D_LANGUAGE : Langue utilisée (en AF_INIT uniquement)

(*) Données non transmises si l'AFSEC+ est en mode 'résident'

6.2.2 Message IC_INIT (0x81)

Message avec les données possibles vers l'AFSEC+ :

- D_PROTOCOL_VERSION : Version du protocole AFSEC+/ICom utilisée par l'ICom
- D_ICOM_VERSION : Version du logiciel ICom de l'interface

Message avec les données possibles vers la Télécommande :

- D_PROTOCOL_VERSION : Version x.y.z du protocole AFSEC+/ICom utilisée par l'AFSEC+
- D_APPLI_NUMBER (*) : Numéro logiciel application de l'AFSEC+
- D_APPLI_VERSION (*) : Numéro de version du logiciel application de l'AFSEC+
- D_APPLI_CONFIG (*) : Identifiant de la configuration 'usine' de l'application AFSEC+
- D_LANGUAGE : Langue utilisée

6.3 Message MENU (MC+ Uniquement)

Cette conversation est initiée par l'AFSEC+ pour permettre à l'ICom d'accéder aux ressources « afficheur » et « boutons poussoirs » de l'AFSEC+ pour y dérouler une séquence de menus avec l'utilisateur.

- AF_MENU : Message de l'AFSEC+ -> ICom pour l'inviter à proposer un menu à l'utilisateur
- ACK : Réponse de l'ICom pour indiquer qu'il n'y a pas (ou plus) de menu à proposer
- IC_MENU : Message réponse de l'ICom -> AFSEC+ pour définir le menu à présenter à l'utilisateur et les actions possibles

Une conversation MENU peut être interrompue par l'AFSEC+ à tout moment, même si l'ICom n'a pas demandé la fin des menus par une réponse NACK.

6.3.1 Usage des menus

L'AFSEC+ propose à l'iCom d'engager une conversation MENU par un message AF_MENU en proposant par défaut le menu référencé par l'ID numéro 1 (D_MENU_ID) et en indiquant le mode courant de fonctionnement de l'AFSEC+ (D_MODE_AFSEC).

L'ICom peut renoncer à cette conversation en répondant NAK.


Sinon, l'iCom répond par un message IC_MENU pour définir le « menu ».
Ou L'AFSEC répond à un message AF_MENU

Un menu est défini par:

- Un ID du menu (D_MENU_ID)
- Eventuellement, le texte de 6 caractères max pour l'afficheur AFSEC+ (D_MENU_SHORT_DISPLAY)
- Eventuellement, la présentation sur le prompteur d'un texte de 20 caractères max (D_MENU_LONG_DISPLAY)
- Eventuellement, les pictogrammes à allumer (D_MENU_PICTOS)

L'AFSEC+ peut prendre en charge plusieurs types de menus :

- Un **menu simple d'une action sur les boutons poussoirs** OK, MENU ou CLEAR :
 - L'ICom spécifie les BP à prendre en compte et les ID menus qui leur sont associés (D_MENU_ID_ON_BP_OK, D_MENU_ON_BP_MENU et/ou D_MENU_ON_BP_CLEAR)

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 22/52
	Service Développement	

- Un **menu de sélection d'un choix dans une liste** :
 - L'ICom spécifie la valeur initiale (D_MENU_VALUE_INIT), la liste des choix possibles (D_MENU_CHOICE_LIST) et – optionnellement - l'ID du menu à invoquer ensuite (D_MENU_ID_ON_BP_OK)
- Un **menu de saisie** :
 - L'ICom spécifie la valeur initiale (D_MENU_VALUE_INIT) son masque de saisie (D_MENU_INPUT_MASK) et – optionnellement - l'ID menu à suivre (D_MENU_ID_ON_BP_OK)

L'AFSEC+ détermine le type de menu selon :

Si présence du champ D_MENU_CHOICE_LIST Alors c'est un menu de type sélection d'un choix dans cette liste

Sinon Si présence du champ D_MENU_INPUT_MASK Alors c'est un menu de type sélection d'un choix dans cette liste

Sinon c'est un menu simple

Nota : Pour des raisons d'implémentation, la valeur 0xFFFF ne doit pas être utilisée pour un D_MENU_ID.

Pendant toute la durée de gestion du menu par l'AFSEC+, la conversation est entretenue par un échange AF_MENU qui contient D_MENU_ID_IN_PROGRESS pour rappeler l'ID_MENU en cours.

L'ICom répond alors :

- Par ACK pour que la gestion du menu en cours se poursuive (signe de vie).
- Il est également possible ici pour l'ICom de retourner des nouvelles valeurs pour les champs possiblement 'dynamiques' : D_MENU_SHORT_DISPLAY, D_MENU_LONG_DISPLAY ou D_MENU_PICTOS dans un message IC_MENU à la place de ACK.
- L'ICom peut également répondre par NAK pour stopper le menu en cours et mettre fin à la conversation.

Lorsque le menu est validé par l'utilisateur, l'AFSEC+ envoie un message AF_MENU avec l'ID du menu à suivre (D_MENU_ID) s'il a été défini précédemment par l'ICom (D_MENU_ID_ON_BP_xxx selon le type de menu).


Pour les menus de type sélection dans une liste ou de saisie, l'AFSEC+ transmet D_MENU_USER_INPUT qui correspond au choix ou à la saisie effectuée par l'utilisateur.

Ce message est similaire à celui utilisé pour engager la conversation.

L'ICom poursuit donc comme décrit ci-dessus.

6.3.2 Message AF_MENU (0x02)

Message avec les données suivantes selon le cas d'utilisation :

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 23/52
	Service Développement	

Cas d'utilisation	D_MENU_ID	D_MODE_AFSEC , D_LANGUAGE	D_MENU_ID_IN_PROGRES	D_MENU_USER_INPUT
Début d'une conversation MENU	= 1	= Mode courant de fonctionnement de l'AFSEC+ et langue choisie par l'utilisateur	-	-
Menu en cours par l'utilisateur	-	-	= MENU_ID du menu en cours	-
Choix selon bouton poussoir	= ID du menu ICom	-	-	-
Choix dans une liste	= ID du menu ICom	-	-	= Choix utilisateur
Saisie	= ID du menu ICom	-	-	= Saisie utilisateur

6.3.3 Message IC_MENU (0x82)

Message avec les données suivantes selon le cas d'utilisation :

- Terminer la conversation MENU : Ne pas utiliser la réponse longue IC_MENU mais utiliser la réponse courte NAK
- Entretenir la conversation pendant que le menu est en cours avec l'utilisateur : ACK
- Entretenir la conversation sur le menu 'dynamique' en cours en indiquant les éventuelles nouvelles valeurs pour :
 - D_MENU_SHORT_DISPLAY : Optionnel
 - D_MENU_LONG_DISPLAY : Optionnel
 - D_MENU_PICTOS : Optionnel

Nota : Toute valeur pour un autre champ que ci-dessus dans cette réponse est ignorée
- Proposer un choix selon les boutons poussoirs :
 - D_MENU_ID : Obligatoire
 - D_MENU_SHORT_DISPLAY : Optionnel
 - D_MENU_LONG_DISPLAY : Optionnel
 - D_MENU_PICTOS : Optionnel
 - D_MENU_ID_ON_BP_OK : Optionnel
 - D_MENU_ID_ON_BP_MENU : Optionnel
 - D_MENU_ID_ON_BP_CLEAR : Optionnel

Au moins un parmi SHORT/LONG DISPLAY et au moins un parmi BP_OK, MENU ou CLEAR fait sens
- Proposer un choix dans une liste de choix :
 - D_MENU_ID : Obligatoire

- D_MENU_SHORT_DISPLAY : Optionnel
- D_MENU_LONG_DISPLAY : Obligatoire
- D_MENU_PICTOS : Optionnel
- D_MENU_ID_ON_BP_OK : Optionnel
- D_MENU_VALUE_INIT : Obligatoire
- D_MENU_CHOICE_LIST : Obligatoire

○ Proposer une saisie :

- D_MENU_ID : Obligatoire
- D_MENU_SHORT_DISPLAY : Optionnel (*)
- D_MENU_LONG_DISPLAY : Obligatoire
- D_MENU_PICTOS : Optionnel
- D_MENU_ID_ON_BP_OK : Optionnel
- D_MENU_VALUE_INIT : Obligatoire (**)
- D_MENU_INPUT_MASK : Obligatoire (**)(***)

(*) Selon l'implémentation, le D_MENU_SHORT_DISPLAY n'est pas affiché pendant la saisie

(**) La longueur max de la chaîne en saisie est limitée à 20 caractères.

(***) Voir les annexes pour les masques de saisie possibles

Nota : L'ICom ne peut pas changer de type de menu pendant qu'un menu est en cours.

6.3.4 Exemples de conversations

Conversation MENU rejetée par l'ICom :

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_MENU avec D_MENU_ID = 1 et D_MODE_AFSEC	L'AFSEC+ propose à l'ICom de débiter une conversation MENU
ICom -> AFSEC+	NAK	L'ICom ne souhaite pas engager une conversation MENU
AFSEC+ -> ICom	AF_ALIVE	Fin de la conversation par l'AFSEC+. Tout message autre que AF_MENU envoyé ici met fin à la conversation et en débute éventuellement une autre

Conversation simple MENU :

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_MENU avec D_MENU_ID = 1 et D_MODE_AFSEC	L'AFSEC+ propose à l'ICom de débiter une conversation MENU
ICom -> AFSEC+	IC_MENU avec : D_MENU_ID = 2, D_MENU_LONG_DISPLAY = "texte", D_MENU_ON_BP_OK = 3	L'ICom propose un menu simple avec un message à afficher pour lequel on ne peut qu'appuyer sur le bouton poussoir OK

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_MENU avec D_MENU_ID_IN_PROGRESS = 2	L'AFSEC+ entretient la conversation pendant que le menu est proposé à l'utilisateur. Ce message est envoyé périodiquement jusqu'à ce que l'utilisateur réalise une action (ici l'appui sur le bouton poussoir OK)
ICom -> AFSEC+	ACK	L'ICom acquitte pour maintenir la conversation et donc le menu proposé à l'utilisateur
AFSEC+ -> ICom	AF_MENU avec D_MENU_ID = 3 et D_MODE_AFSEC	L'AFSEC+ indique que l'opérateur à réaliser une action (ici la seule action possible qui mène au nouveau menu ID = 3)
ICom -> AFSEC+	NAK	L'ICom ne souhaite pas poursuivre cette conversation MENU
AFSEC+ -> ICom	AF_ALIVE	Fin de la conversation par l'AFSEC+. Tout message autre que AF_MENU envoyé ici met fin à la conversation et en débute éventuellement une autre

6.4 RF MENU (RF uniquement)

L'échange des menus entre la RCT5 et l'AFSEC est très proche des échanges DATA_IN et DATA_OUT.

Cette conversion est initiée par la RTC5 pour permettre à l'AFSEC d'accéder aux ressources « afficheur » et « boutons poussoirs » de celui-ci.

La fonctionnalité secondaire de cette conversation est de synchroniser les tables d'index de libelles en RAM

- TL_RFMENU : Message de RCT5 → iCom → AFSEC pour l'inviter à proposer un menu à l'utilisateur (la partie iCom → AFSEC est basé sur l'échange DATA_IN)
- AF_RFMENU : Message réponse de l'AFSEC+ → iCom → RCT5 pour définir le menu à présenter à l'utilisateur et les actions possibles l'utilisateur (la partie AFSEC → iCom est basé sur l'échange DATA_OUT)

6.4.1 Message TL_RFMENU (0x0A)

Les données possibles dans le message TL_RFMENU sont :

- D_DATA_ZONE : Zone des données à suivre (optionel)
- D_DATA_TAG : Identification de la donnée transmise
- D_DATA_VALUE : Valeur de la donnée

Les réponses possibles de l'ICom sont :

- NACK : Réponse si l'afsec n'a pas retourné les infos nécessaires pour maintenir la conversation d mode menu distant

- **TL_RFMENU** : Réponse longue de l'ICom pour indiquer la réponse du menu en en cours.

6.5 **Message DATA**

Cette conversation est engagée par l'ASEC+, la télécommande ou l'ICom pour échanger des données applicatives.

Les messages 'DATA' permettent d'échanger des données 'atomiques' référencées par un 'tag'. Depuis la révision 0.0.3 du protocole, il existe également les message 'PACK' pour échanger des zones mémoire de données 'bruts' (typiquement des 'portions' de tables MODBUS)

Les données applicatives sont élaborées ou utilisées par l'application AFSEC+.

Un paramètre de la configuration métrologique, un item d'un enregistrement du journal métrologique ou un ordre d'autorisation de chargement sont des exemples de données applicatives.

Une conversion DATA_OUT concerne l'envoi d'un ensemble de données de l'AFSEC+ vers ICom [Les données détenues en interne par l'AFSEC+ et qui sont destinées à être externalisées].

Inversement, une conversation DATA_IN est l'envoi d'un ensemble de données de l'ICom vers l'AFSEC+ [Des données de provenance externes et qui sont transmises à l'AFSEC+].

Les conversations DATA_IN et DATA_OUT sont différenciées et ne peuvent pas avoir lieu simultanément.

Dans une conversation DATA, les données échangées à un instant donné font partie d'une zone.

La table des zones en annexe définit les différentes zones possibles.


La zone est une information contextuelle de la conversation DATA. Ce qui signifie qu'en absence de précision dans une transaction, c'est la dernière zone référencée qui est celle en cours.

Certaines zones correspondent physiquement à des tables d'enregistrements. Typiquement, le journal métrologique des résultats de mesure ou le journal des événements.

Dans ce cas, la donnée est associée à un index relatif à l'enregistrement dans ce journal appelé 'Table Index'.

L'index est techniquement une valeur numérique (sur 64 bits) mais sa construction identifie de manière non ambiguë un enregistrement dans le temps (date/heure ou date/numéro d'ordre dans le jour – Voir les détails en annexe).

La SR DEV 0004 définit un TAG particulier qui signifie 'END_OF_RECORD' (et dont la valeur est sans importance).

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 27/52
	Service Développement	

Lors des transmissions du contenu d'un enregistrement dans une zone indexées, Il convient d'envoyer un dernier TAG='END_OF_RECORD'/VALUE='peut-importe' pour indiquer qu'il n'y a pas d'autre information à suivre concernant cet enregistrement.

Nota : Ici, '*il convient*' d'envoyer ce TAG='END_OF_RECORD' signifie que le destinataire doit pouvoir gérer des envois de records qui n'intègrent pas de 'END_OF_RECORD' (en implémentant un timeout, par exemple).

L'index table n'est pas précisé pour les données qui n'appartiennent pas à une zone organisée en table.

Les données transmises sont 'taguées' selon la spécification **SR DEV 0004 {Protocole µConfig}** : TAG (16 bits) + 3 indices (3 x 8 bits), soit 5 octets au total.

Une donnée transmise est donc caractérisée par l'ensemble suivant :

- **Zone** : Numéro de la zone de la donnée. Contextuel : La zone définie reste valide pour toutes les données suivantes transmises dans la même conversation jusqu'à spécification d'une nouvelle zone
- **Index Table** : Optionnel pour les zones organisées en table. Contextuel : L'index défini reste valide pour toutes les données suivantes transmises dans la même conversation jusqu'à spécification d'un autre index

Nota : Par convention, la valeur d'un index table par défaut est 0 (contextuel et pour les zones qui ne sont pas organisées en table)

- **Tag** : Tag et 3 indices (soit 5 caractères au total) selon SR DEV 0004 de la donnée
- **Valeur** : Valeur de la donnée

6.5.1 Message AF_DATA_OUT ou TL_DATA_OUT (0x03)


Cette conversation est initiée par l'AFSEC+ pour transmettre une ou plusieurs données de l'AFSEC+ vers l'ICom.

La conversation est maintenue tant que l'AFSEC+ envoie des AF_DATA_OUT à l'ICom.

La conversation prend normalement fin avec un message AF_ALIVE.

Les données possibles dans le message AF_DATA_OUT sont :

- D_DATA_ZONE : Zone des données à suivre
- D_DATA_TABLE_INDEX : Utilisé pour les zones qui sont des tables d'enregistrements

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 28/52
	Service Développement	

- D_DATA_TAG : Identification de la donnée transmise
- ~~D_DATA_USAGE : Non utilisé, usage futur~~
- D_DATA_VALUE : Valeur de la donnée
- D_DATA_TAG='END_OF_RECORD' (voir SR DEV 0004) et D_DATA_VALUE=True : Pour indiquer la fin de tous les TAG/VALUE d'un enregistrement d'une table

Les réponses possibles de l'ICom sont :

- ACK : Réponse courte pour acquitter la bonne réception du message. L'AFSEC+ poursuit la conversation
- IC_DATA_OUT : Réponse longue de l'ICom pour indiquer un souci dans le transfert. Selon le souci signalé, l'AFSEC+ décide de poursuivre ou de stopper la conversation

Nota : L'ICom ne peut pas répondre par un IC_DATA_IN à un AF_DATA_OUT de l'AFSEC+. Il s'agit de 2 conversations différentes.

6.5.2 Message IC_DATA_OUT (0x83)

Ce message est une réponse possible de l'ICom à un message AF_DATA_OUT.

Il signale généralement un souci constaté par l'ICom pour inviter l'AFSEC+ à cesser le transfert des données en utilisant D_DATA_ERROR.

Nota : Un message IC_DATA_OUT sans donnée est équivalent à un ACK de la transmission

Nota 2 : Un code d'erreur D_DATA_ERROR = 0 est équivalent à un ACK de la transmission

Nota 3 : Aucun code d'erreur n'oblige l'AFSEC+ à cesser la conversation.

6.5.3 Message IC_DATA_IN (0x84)


Ce message est une réponse possible de l'ICom à un message :

AF_ALIVE : Invite de l'AFSEC+ à transmettre

AF_DATA_IN : Pour suivre d'une conversion DATA_IN en cours

Les données possibles dans ces messages sont les mêmes que pour AF_DATA_OUT et selon les mêmes modalités.

L'AFSEC+ entretient alors la conversation par un AF_DATA_IN pour inviter l'ICom à poursuivre.

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 29/52
	Service Développement	

Si un autre message est émis par l'AFSEC+, la conversion est alors terminée et les données transmises précédemment sont à renvoyer.

6.5.4 Message AF_DATA_IN (0x04)

Ce message est émis par l'AFSEC+ pour inviter l'ICom à poursuivre la conversation DATA_IN en cours (suite à un IC_DATA_IN).

Les données possibles dans ce message sont les mêmes que pour un IC_DATA_OUT et selon les mêmes modalités.

Les réponses possibles de l'ICom sont :

- IC_DATA_IN : Pour poursuivre la conversation DATA_IN
- ACK ou NAK pour terminer la conversion (plus de données à transmettre)

6.5.5 Cinématique des conversations DATA_IN et DATA_OUT


6.5.5.1 Règles de gestion

Pour le bon déroulement des conversations d'échange de données applicatives, les règles de gestion suivantes sont applicables :

1. Tout message n'appartenant pas à l'ensemble décrit ci-dessus rompt la conversation en cours d'un transfert de données applicatives. Il revient alors à l'AFSEC+ d'amorcer une nouvelle conversation par une nouvelle vacation
2. L'ordre dans lequel sont placés les différentes informations TLV dans les trames AF_DATA_OUT et IC_DATA_IN est important
3. Les données D_DATA_ZONE et D_DATA_TABLE_INDEX (si requis) sont les seules qui peuvent être contextuelles. Ce qui signifie qu'il n'est pas nécessaire de les répéter pour une nouvelle donnée applicative à transmettre si leur valeur reste la même que pour la donnée précédente. Par défaut, ces 2 données sont implicitement à 0
4. Le destinataire d'un message enregistre techniquement la donnée transmise dès lors qu'un nouveau couple D_DATA_TAG et D_DATA_VALUE est reçu
5. Si plusieurs D_DATA_TAG sont reçus avant le D_DATA_VALUE, c'est le dernier D_DATA_TAG reçu qui fait foi. Idem dans la situation inverse ou plusieurs D_DATA_VALUE sont reçus avant le D_DATA_TAG

6.5.5.2 Exemples de conversations

Conversation DATA_OUT valide :

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 30/52
	Service Développement	

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_DATA_OUT avec D_DATA_ZONE, D_DATA_TAG et D_DATA_VALUE	L'AFSEC+ débute une conversation DATA_OUT. Le message contient toutes les infos suffisantes pour la transmission d'une donnée applicative (la zone n'est pas un tableau)
ICom -> AFSEC+	ACK	Acquit bonne réception de l'ICom
AFSEC+ -> ICom	AF_ALIVE	Fin de la conversation par l'AFSEC+. Tout message autre que AF_DATA_OUT envoyé ici met fin à la conversation et en débute éventuellement une autre

Conversation DATA_IN valide :

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_ALIVE	Invite à parler de l'AFSEC+
ICom -> AFSEC+	IC_DATA_IN avec D_DATA_ZONE, D_DATA_TAG et D_DATA_VALUE	L'ICom débute une conversation DATA_IN. Le message contient toutes les infos suffisantes pour la transmission d'une donnée applicative (la zone n'est pas un tableau)
AFSEC+ -> ICom	AF_DATA_IN avec D_DATA_ERROR = 0	Acquit bonne réception de l'AFSEC+ pour entretenir la conversation DATA_IN entamée par l'ICom
ICom -> AFSEC+	ACK	L'ICom met fin à la conversation
AFSEC+ -> ICom	AF_ALIVE	L'AFSEC+ peut débiter ici une nouvelle conversation

Conversation plus complexe valide :

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_DATA_OUT avec D_DATA_ZONE1, D_DATA_TAG1, D_DATA_VALUE1, D_DATA_TAG2, D_DATA_VALUE2	L'AFSEC+ débute une conversation DATA_OUT
ICom -> AFSEC+	ACK	Acquit bonne réception de l'ICom qui enregistre dans D_DATA_ZONE1 les données applicatives pour TAG1 = VALUE1 et TAG2 = VALUE2
AFSEC+ -> ICom	AF_DATA_OUT avec D_DATA_VALUE3, D_DATA_TAG3, D_DATA_TAG Ignoré, D_DATA_TAG4, D_DATA_VALUE4, D_DATA_ZONE2, D_DATA_TAG5, D_DATA_VALUE5	L'AFSEC+ poursuit la conversation DATA_OUT
ICom -> AFSEC+	ACK	Acquit bonne réception de l'ICom qui enregistre dans D_DATA_ZONE1 (contexte) les données applicatives pour TAG3 = VALUE3 (TAG3 a été reçu après VALUE3) et TAG4 = VALUE4 (TAG Ignoré n'est pas considéré car TAG4 est reçu ensuite). L'ICom enregistre également TAG5 = VALUE5 dans la ZONE2
AFSEC+ -> ICom	AF_ALIVE	Fin de la conversation par l'AFSEC+

6.5.6 Message AF_DATA_OUT_TABLE_INDEX (0x05) (MC+ uniquement)

Ce message est émis par l'AFSEC+ pour obtenir de l'ICom le premier et le dernier index connus pour une zone organisée en table.

Typiquement, ce message permet à l'AFSEC+ de synchroniser la mise à jour des données applicatives contenues dans une zone organisée en table en obtenant l'index du premier et l'index du dernier enregistrement reçu et mémorisé par l'ICom.

La donnée associée à ce message est :

- D_DATA_ZONE : Zone des données concernée

La seule réponse possible de l'ICom est un message IC_DATA_OUT_TABLE_INDEX.

Nota : Chaque conversation AF_DATA_OUT_TABLE_INDEX / IC_DATA_OUT_TABLE_INDEX ne porte que sur une seule zone de données. On répète la conversation autant de fois que nécessaire pour chaque zone concernée.

6.5.7 Message IC_DATA_OUT_TABLE_INDEX (0x85) (MC+ uniquement)

Ce message est la seule réponse possible de l'ICom à un message AF_DATA_OUT_TABLE_INDEX.


Il contient :

- D_DATA_ZONE : Zone des données demandée dans le message IC_DATA_OUT_TABLE_INDEX
- D_DATA_FIRST_TABLE_INDEX : Avec la valeur de TABLE_INDEX du premier item de la zone qui est détenu par l'ICom (le plus ancien) par un D_DATA_OUT sur cette zone. Par défaut, si la zone n'est pas une zone organisée en table ou si aucun enregistrement n'est détenu par l'ICom pour cette zone, la valeur retournée est 0
- D_DATA_LAST_TABLE_INDEX : Avec la valeur de TABLE_INDEX du dernier item de la zone qui est détenu par l'ICom (le dernier reçu) par un D_DATA_OUT sur cette zone. Par défaut, si la zone n'est pas une zone organisée en table ou si aucun enregistrement n'est détenu par l'ICom pour cette zone, la valeur retournée est 0

Nota : Pour des raisons techniques liées à l'architecture de l'AFSEC+, les données D_DATA_FIRST_TABLE_INDEX et D_DATA_LAST_TABLE_INDEX doivent obligatoirement être encodées au format entier 64bits.

6.6 Message DATA_IN_REQ (RF Uniquement)

Ce message permet de faire une requête à l'ICom pour qu'elle envoie des données de façon récurrente (automatiquement et régulièrement) ou juste une seule fois.

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 32/52
	Service Développement	

- TL_DATA_IN_REQ : Message Télécommande -> ICom pour demander l'envoi d'une donnée récurrente
- IC_DATA_IN_REQ : Message ICom -> Télécommande pour accepter ou refuser la demande d'envoi d'une donnée récurrente
- IC_DATA_IN_RES : Message ICom -> Télécommande pour envoyer (une fois ou périodiquement) la valeur d'une donnée récurrente sur invitation par un TL_ALIVE

Lorsque l'ICom accepte l'envoi de données récurrentes, les valeurs des tags demandés sont transmises par l'ICom par des messages IC_DATA_IN_RES en réponse à une invitation TL_ALIVE de la télécommande.

Le protocole permet de définir 8 tags différents faisant simultanément l'objet d'une demande d'envoi par récurrence.

Pour alléger la bande passante dans les messages IC_DATA_IN_RES, c'est le numéro de slot (de 0 à 7) associé lors de la demande qui est un alias pour ne pas envoyer systématiquement le D_DATA_TAG avec chaque message

L'ICom priorisera et organisera sa gestion des tags avec demande de récurrence pour :

1. Envoyer une valeur qu'après la date passée pour l'échéance de récurrence demandée
2. Envoyer la dernière valeur 'à jour' de cette donnée
 - a. Pas de file d'attente pour envoyer d'anciennes valeurs 'obsolètes' des données
 - b. Pas d'envoi s'il n'y a pas eu de mise à jour de la donnée depuis la dernière récurrence transmise

Charge à la télécommande de cadencer ces demandes de récurrences et ces invitations transmettre via un TL_ALIVE.


6.6.1 Message TL_DATA_IN_REQ (0x08)

Message avec les données possibles :

- D_DATA_ZONE : Zone de la donnée
- D_DATA_TAG : Tag de la donnée
- D_DATA_REQ : Requête de récurrence de la donnée
- D_DATA_REQ_SLOTx : Identification du slot (de D_DATA_REQ_SLOT0 à D_DATA_REQ_SLOT7) pour cette demande

6.6.2 Message IC_DATA_IN_REQ (0x88)

Réponse à TL_DATA_IN_REQ :

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 33/52
	Service Développement	

- ACK : Requête validée
- NACK : Donnée non disponible (ou fonctionnalité non supportée)

6.6.3 Message IC_DATA_IN_RES (0x89)

Message envoyé par l'ICom sur invitation par un AF_ALIVE pour transmettre la valeur d'une donnée sous récurrence :

- D_DATA_REQ_SLOTx : Identification du slot associé à la demande
- D_DATA_VALUE : Valeur de la donnée associée

Ce message ne doit pas contenir plus de 2 couples (D_DATA_REQ_SLOTx, D_DATA_VALUE) : Préservation de la bande passante.

Exemple de conversation DATA_IN_REQ valide :

Sens du message	Contenu du message	Note
Télécommande -> ICom	TL_DATA_IN_REQ avec : D_DATA_ZONE =4 D_DATA_TAG = 0x1234000000 D_DATA_REQ = 0x0108 D_DATA_REQ_SLOT4	La télécommande demande à l'ICom de lui envoyer la valeur du tag 0x1234:00:00:00 en zone 4 chaque seconde pendant 8 secondes et de référencer cet envoi par le slot#4
ICom -> Télécommande	ACK	L'ICom accepte
Télécommande -> ICom	TC_ALIVE	La télécommande invite l'ICom à parler
ICom -> Télécommande	IC_DATA_RES avec : D_DATA_REQ_SLOT4 D_DATA_VALUE=123	L'ICom transmet la dernière valeur à jour du tag associé au slot#4

6.7 Messages PACK (0x0B, 0x0C, 0x8B et 0x8C)

Une conversion PACK permet d'échanger des portions de données 'bruts' qui n'ont de signification que pour l'expéditeur et le destinataire (client externe de l'ICOM et application AFSEC+).


Ces messages sont destinés à optimiser des échanges de tables MODBUS de 256 mots max. dont le contenu est spécialisé entre un client MODBUS ou MODBUS/TCP de l'ICOM et l'application AFSEC+.

Une conversion PACK_OUT concerne une portion de données de la table de lecture générée par l'application AFSEC+ et à disposition d'un client MODBUS ou MODBUS/TCP de l'ICOM. Il s'agit alors d'un flux de données dans le sens AF -> IC.

Une conversion PACK_IN concerne une portion de données de la table d'écriture mise à jour par un client MODBUS ou MODBUS/TCP de l'ICOM à destination de l'application AFSEC+. Il s'agit alors d'un flux de données dans le sens IC -> AF.

La taille maximum des tables échangées est de 256 mots (512 octets).

Cette taille dépasse les possibilités du protocole en un seul message.

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 34/52
	Service Développement	

Le type de données 'D_PACK_PAYLOAD' est utilisé pour échanger une 'partie' de table à synchroniser dans les messages.

Un message PACK_IN ou PACK_OUT peut intégrer plusieurs parties 'D_PACK_PAYLOAD' dans son contenu.

Ce découpage se fait en numérotant les paquets de données qui sont transmis au cours d'une conversation pour synchroniser l'ensemble de la table.

Par convention, on découpe par paquets d'au plus 2 + 64 octets.

Par convention, le type de données dans ce paquet est un String(66) ou une chaîne de longueur plus courte.

Par convention, on intègre un nombre pair d'octets dans chaque paquet.

Dans l'absolu, rien n'interdit un paquet contenant 125 octets (la taille max. d'une chaîne étant 127).

Le nombre total de paquets ne peut pas dépasser 15 paquets.

Les paquets doivent être numérotés de 1 à N (N étant le nombre total de paquet annoncé) dans l'ordre au sein d'un même message et dans la continuité des conversations. Une rupture de séquence ou une modification du nombre total de paquets n'est pas conforme.

Dans les cas les plus défavorables, il peut y avoir 3 données 'D_PACK_PAYLOAD' de 64 octets à synchroniser dans un message PACK_IN ou PACK_OUT (204 octets au total pour transporter 192 octets 'utiles').

Et il peut y avoir jusqu'à 3 paquets à transmettre ($3 \times 192 > 512$) pour synchroniser l'ensemble d'une table de 256 mots.

Ainsi, les messages PACK_IN et PACK_OUT intègrent nativement :

- **T (tag)**
 - AF_PACK_OUT (0x0B) : L'AFSEC+ communique des données du 'data-pack-out' à l'ICOM (à l'initiative de l'AFSEC+)
 - IC_PACK_IN (0x8C) : L'ICOM communique des données du 'data-pack-in' à l'AFSEC+ (sur invite à parler de l'AFSEC+)
- **L (longueur)** = nombre d'octets des D_PACK_PAYLOAD à suivre qui peuvent être répété plusieurs fois dans le message :
 - **T (tag)** : D_PACK_PAYLOAD
 - **L (type de donnée)** : String(2 + n)
 - **PACK_NUMBER** 2 x quartet sur un UI_8: Numéro du paquet (base 1) sur le nombre total de paquets de cette transaction. Exemple 0x12 signifie qu'il s'agit du premier paquet sur un total de 2
 - **ADDR_BASE UI_8** : Adresse de base ****en mots**** (0-255)

- Données 'raw binaire' du pack (max. théorique de 125 octets mais dans la pratique, on limite à 64 octets (32 mots)
Il est d'usage que ce nombre d'octets soit une valeur paire puisqu'il s'agit de valeurs pour une table de mots.

6.7.1 Message AF_PACK_OUT (0x0B) et IC_PACK_OUT (0x8B)

L'AFSEC+ envoie des messages AF_PACK_OUT avec des D_PACK_PAYLOAD à sa convenance et attend une réponse de l'ICOM :

- ACK ou IC_PACK_OUT (sans donnée) pour accuser réception des données
- NACK pour redemander la transmission de ce paquet
- Tout autre réponse est traitée comme un NACK par l'AFSEC+

6.7.2 Message IC_PACK_IN (0x8C) et AF_PACK_IN (0x0C)

L'ICOM envoie des messages IC_PACK_IN avec des D_PACK_PAYLOAD à sa convenance sur invitation à parler par l'AFSEC+ (ALIVE).

La réponse de l'AFSEC+ peut être :

- AF_PACK_IN (sans donnée) pour accuser réception du paquet et inviter l'ICOM à transmettre le paquet suivant dans la même conversation
- ACK pour accuser réception des données et terminer la conversation. L'ICOM doit alors attendre une nouvelle invite à une conversation par l'AFSEC+ pour poursuivre la transaction si nécessaire
- NACK pour ne pas accuser réception des données et terminer la conversation

6.7.3 Exemples de conversations

Exemple de conversation PACK_OUT valide :

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_PACK_OUT avec D_PACK_PAYLOAD = 0x12, 0x00, 0x01, 0x02, 0x03, 0x04	L'AFSEC+ débute une conversation PACK_OUT. Il s'agit du paquet 1 sur 2 pour une mise à jour de la table de lecture à l'adresse mot 0 avec les valeurs : mot0 = 0x0102 et mot1 = 0x0304
ICom -> AFSEC+	ACK (ou IC_PACK_OUT sans donnée)	Acquit bonne réception de l'ICom
AFSEC+ -> ICom	AF_PACK_OUT avec D_PACK_PAYLOAD = 0x22, 0x02, 0x05, 0x06, 0x07, 0x08	L'AFSEC+ envoie le second et dernier paquet pour une mise à jour de la table de lecture à l'adresse mot 2 avec les valeurs : mot 2 = 0x0506 et mot3 = 0x0708
ICom -> AFSEC+	ACK (ou IC_PACK_OUT sans données)	Acquit bonne réception de l'ICom qui peut maintenant exposer le contenu d'une table de lecture mise à jour entre le mot 0 et le mot 3 = 0x0102030405060708 (Fin implicite de cette conversation)
AFSEC+ -> ICom	AF_ALIVE	L'AFSEC+ invite l'ICOM à débiter une autre conversation

Exemple de conversation PACK_IN valide :

Sens du message	Contenu du message	Note
AFSEC+ -> ICom	AF_ALIVE	Invite à parler de l'AFSEC+
ICom -> AFSEC+	IC_PACK_IN avec D_PACK_PAYLOAD = 0x12, 0x10, 0x01, 0x02, 0x03, 0x04	L'ICom débute une conversation PACK_IN. Il s'agit du paquet 1 sur 2 pour une mise à jour de la table d'écriture à l'adresse mot 0x0010 avec les valeurs : mot10 = 0x0102 et mot11 = 0x0304
AFSEC+ -> ICom	AF_DATA_IN (sans donnée)	Acquit bonne réception de l'AFSEC+ pour entretenir la conversation DATA_IN entamée par l'ICom
ICom -> AFSEC+	IC_PACK_IN avec D_PACK_PAYLOAD = 0x22, 0x12, 0x05, 0x06, 0x07, 0x08	L'ICOM+ envoie le second et dernier paquet pour une mise à jour de la table d'écriture à l'adresse mot 0x22 avec les valeurs : mot 22 = 0x0506 et mot23 = 0x0708
AFSEC+ -> ICom	ACK	L'AFSEC+ acquitte la réception de ce second paquet et met fin à la conversation
AFSEC+ -> ICom	AF_ALIVE	L'AFSEC+ débute ici une nouvelle conversation Ici, elle propose à l'ICOM de débiter une autre conversation

6.8 Message DOWNLOAD (MC+ uniquement)

Une conversation 'DOWNLOAD' permet de télécharger du code dans la FLASH de l'AFSEC+.

Il peut s'agir de 3 sections dédiées de l'AFSEC+ :

1. Un programme d'application
2. Un lot de tickets
3. Un catalogue de traduction

Nota : Le téléchargement d'un résident est exclu.

Ce téléchargement est à l'initiative de l'ICom.

Les données téléchargées sont les mêmes que celles contenues dans un fichier qui pourrait être téléchargé par l'outil TéléConf standard.

6.8.1 Message IC_DOWNLOAD (0x86)

Ce message est une réponse possible de l'ICom à un message

AF_ALIVE : Invite de l'AFSEC+ à transmettre

AF_DOWNLOAD : Pour suivre d'une conversion DOWNLOAD en cours

Lorsque l'ICom répond à AF_ALIVE par un message IC_DOWNLOAD, cela indique son intention de réaliser le téléchargement d'un fichier. Le message doit alors définir D_DOWNLOAD_SECTION pour indiquer quelles données vont être transférées.

Les données D_DOWNLOAD_NAME et D_DOWNLOAD_NB_RECORDS peuvent également accompagner ce message.

Si le téléchargement est accepté par l'AFSEC+, les messages IC_DOWNLOAD suivants contiennent un (ou plusieurs) D_DOWNLOAD_RECORD contenant une ligne du fichier à télécharger au format S-Record (également appelé S1-S9 ou s19) de MOTOROLA ([Voir le détail ici](#)). Il n'est pas nécessaire de marquer la fin de la ligne par un CR et/ou un LF.

Il n'est pas nécessaire de répéter le D_DOWNLOAD_SECTION en cours.

Après que la dernière ligne du fichier soit transmise, l'ICom envoie un message contenant D_DOWNLOAD_END (peu importe la valeur) pour indiquer qu'il n'y a plus d'enregistrement à suivre.

L'ICom met fin à la conversation en répondant par ACK au dernier message de compte rendu de téléchargement AF_DOWNLOAD de l'AFSEC+.

Pendant toute la conversation, si l'AFSEC+ répond par un status indiquant une erreur, l'ICom répond par ACK pour mettre fin à la conversation.

6.8.2 AF_DOWNLOAD (0x06)

Ce message est envoyé par l'AFSEC+ pour entretenir une conversion de téléchargement entamée par l'ICom ou pour lui signaler un problème de téléchargement.

La donnée D_DOWNLOAD_STATUS est transmise avec ce message.

En absence (message AF_DOWNLOAD vide de toute donnée), cela signifie que tout est correct (STATUS=OK) et que l'ICom peut poursuivre la conversion de téléchargement.


6.9 Message SYNC (*RF Uniquement*)

Ce message permet à l'ICom et à la télécommande de s'appairer et d'échanger la clé de chiffrement :

- TL_SYNC : Message Télécommande -> ICom
- IC_SYNC : Message ICom -> Télécommande

Tant qu'aucun appairage correct n'est réalisé par cette conversation TL_SYNC / IC_SYNC, toutes les autres conversions entre la télécommande et l'ICom sont impossibles.

Techniquement, cela correspond avec une phase de synchronisation engagée par l'utilisateur avec la télécommande et avec l'ICom et pendant laquelle seuls des messages TC_SYNC et IC_SYNC peuvent être échangés et seulement ceux-ci.

	SR DEV 0006 G du 21.09.2023	Page 38/52
	Protocole AFSEC+/Télécommande Alma vs iCom Service Développement	

Pendant cette phase, le chiffrement AES 256 des données et la reconnaissance du `TypeID` ne sont pas applicables (Voir §5.2).

Les messages SYNC ne sont ensuite jamais utilisés pour toutes les conversations entre la télécommande et l'ICom en utilisation normale.

A l'issue, la télécommande et l'ICom se sont synchronisés et ont toutes les deux mémorisés de façon pérenne :

- `D_CIPHER_KEY` : C'est la clef AES 256 bits qui sera utilisées pour chiffrer toutes les données échangées
- `D_SN_PERIPH télécommande` : C'est le `TypeID` (16 bits avec le bit de poids fort à 1) qui apparaîtra pour toutes les requêtes de la télécommande
- `D_SN_PERIPH ICom` : C'est le `TypeID` (16 bits avec le bit de poids fort à 0) qui apparaîtra pour toutes les réponses de l'ICom

6.9.1 Message TL_SYNC (0x07)

Message avec les données possibles :

- `D_SN_PERIPH` : Numéro de série du périphérique télécommande
- `D_CIPHER_KEY`: Clé générée de façon aléatoire durant la synchronisation avec l'ICom

6.9.2 Message IC_SYNC (0x87)

Message avec les données possibles vers la télécommande :

- `D_SN_PERIPH` : Numéro de série de l'ICom si la synchronisation a été acceptée.
- `NACK` : Echec de synchronisation (Si l'ICom est déjà synchronisé ou qu'elle n'est pas en mode « Sync en cours »)


6.10 Message TEST

Ce message peut être envoyé par l'AFSEC+ ou par l'ICom pour des besoins de tests. Il n'est normalement pas utilisé pour les implémentations en production mais n'est pas interdit.

- `AF_TEST` : Message de l'AFSEC+ -> ICom
- `IC_TEST` : Message de l'ICom -> AFSEC+

6.10.1 Message AF_TEST (0x7F)

Message avec les données possibles :

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 39/52
	Service Développement	

- D_TEST_NB_REQS : Nombre de requêtes de TEST
- D_TEST_NB_REPS : Nombre de réponses reçues à des requêtes de TEST

Les réponses possibles de l'ICom :

- ACK
- IC_TEST

6.10.2 Message IC_TEST (0xFF)

Message avec les mêmes données possibles que pour AF_TEST.

Ce message est une réponse possible à AF_TEST.

Ce message est également possible à l'initiative de l'ICom ; Auquel cas, l'AFSEC+ répond par AF_TEST et s'attend ensuite à un ACK de l'ICom.

6.10.3 Cinématique des conversations de TEST

A l'initiative de l'AFSEC+ :

- Envoi d'AF_TEST par l'AFSEC+
- Réponse IC_TEST de l'ICom
- Autre conversation ensuite

A l'initiative de l'ICom :

- Envoi d'AF_ALIVE par l'AFSEC+ (invitation à engager une conversation)
- Réponse IC_TEST de l'ICom
- Envoi d'AF_TEST par l'AFSEC+
- Réponse ACK de l'ICom

7 ANNEXES

7.1 Codage de la longueur et format d'une donnée TLV

Longueur et Format	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Codage Hexa
<i>String(N)</i>	String=1	Nombre de caractères (de 1 à 127 octets) dans la chaîne							0x81 à 0xFF
<i>Double flottant IEEE</i>	0	Float=1	Signé=1	0	Nombre d'octets=8				0x68
<i>Flottant IEEE</i>	0	Float=1	Signé=1	0	Nombre d'octets=4				0x64
<i>Entier long long non signé</i>	0	0	0	0	Nombre d'octets=8				0x08
<i>Entier long signé</i>	0	0	Signé=1	0	Nombre d'octets=4				0x44
<i>Entier long non signé</i>	0	0	0	0	Nombre d'octets=4				0x04
<i>Entier signé</i>	0	0	Signé=1	0	Nombre d'octets=2				0x42
<i>Entier non signé</i>	0	0	0	0	Nombre d'octets=2				0x02
<i>Char signé</i>	0	0	Signé=1	0	Nombre d'octet = 1				0x41
<i>Char non signé</i>	0	0	0	0	Nombre d'octet=1				0x01
<i>Booléen</i>	0	0	0	Bool=1	Nombre d'octet=1				0x11
<i>Aucune donnée</i>	0	0	0	0	Nombre d'octet=0				0x00

7.2 Tags des données internes (D xxxx)

Symbole	TAG	Format	Domaine	Description
D_TAG_NONE	0x00	(sans objet)	(sans objet)	Utilisé pour 'padder' une liste de valeurs dans un message (voir les particularités pour une communication RF)
D_PROTOCOL_VERSION	0x01	10000 * version + 100 * révision + édition	0..99999	Version du protocole AFSEC+/iCom
D_ICOM_VERSION	0x02	Idem D_RESIDENT_VERSION		Version du programme de l'iCom
D_RESIDENT_VERSION	0x03	Idem D_PROTOCOL_VERSION		Version x.y.z du programme résident de l'AFSEC+ avec x=version, y=révision et z=édition
D_APPLI_NUMBER	0x04	Entier <u>signé</u>	- 99999..99999	Numéro d'identification du logiciel applicatif
D_APPLI_VERSION	0x05	Idem D_RESIDENT_VERSION		Version du programme application AFSEC+
D_APPLI_CONFIG	0x06	String		Identifiant de la configuration 'usine' de l'application AFSEC+
D_MODE_AFSEC	0x07	Numérique	Enumération	Mode actuel de fonctionnement de l'AFSEC+ : 0 : En mode résident 1 : Application en mode métrologique 2 : Application en mode superviseur 3 : Application dans un autre mode (utilisateur)
D_LANGUAGE	0x08	String2	ISO 639-1	Langue utilisée. 'en' par défaut

Symbole	TAG	Format	Domaine	Description
D_MENU_ID	0x10	Numérique	0..0xFFFE	Identification du menu en cours : Par convention : ID = 1 : Menu par défaut ID = 0 : Menu que l'iCom reconnaît comme la fin de sa conversation ID = 0xFFFF : A ne pas utiliser
D_MENU_ID_IN_PROGRESS	0x11	Idem D_MENU_ID		Rappel du menu en cours pendant sa réalisation (maintien de la conversation)
D_MENU_SHORT_DISPLAY	0x12	String 6	0..6 caractères	Affichage d'un menu (partie du haut). Rien, par défaut
D_MENU_LONG_DISPLAY	0x13	String 20	0..20 caractères	Affichage d'un item sur le prompteur. Rien par défaut
D_MENU_PICTOS	0x14	UI_32	32 bits	Combinaison de bits pour allumer un ou plusieurs pictogrammes sur l'afficheur (Par défaut, aucun pictogramme) Voir table de masques dans tableau à suivre
D_MENU_ID_ON_BP_OK	0x15	Idem D_MENU_ID		ID_MENU si choix du BP OK
D_MENU_ID_ON_BP_MENU	0x16	Idem D_MENU_ID		ID_MENU si choix du BP MENU
D_MENU_ID_ON_BP_CLEAR	0x17	Idem D_MENU_ID		ID_MENU si choix du BP CLEAR
D_MENU_VALUE_INIT	0x18	String	0..20 caractères	Valeur initiale pour une saisie (voir le masque de saisie associé)
D_MENU_CHOICE_LIST	0x19	N Strings séparées par une virgule	0..100 caractères	Liste de choix pour un menu de sélection. Chaque choix est un texte. Les différents choix possibles sont séparés par le caractère ',' (ascii=124)
D_MENU_INPUT_MASK	0x1A	String	1..20 (*) caractères (*) <i>Parfois limité à 12 cars selon implémentation</i>	Masque pour la saisie d'un item (voir la valeur initiale de saisie associée) : '+' : Saisie d'un signe '9' : Saisie d'un chiffre 'Z' : Saisie d'une 'printable' (lettre, chiffre, espace, ...) '#' ou autre que ci-dessus : Non éditable
D_MENU_USER_INPUT	0x1B	String	1..20 caractères	Résultat d'une sélection (D_MENU_CHOICE_LIST) ou d'une saisie (D_MENU_INPUT_MASK) par l'utilisateur

Symbole	TAG	Format	Domaine	Description
D_DATA_ERROR	0x30	Numérique	0..99	Codification d'une erreur de transmission d'une donnée applicative. Voir la codification dans la table en annexe
D_DATA_ZONE	0x31	Numérique	0..MAX_ZONES	Numéro de la zone d'une donnée applicative (contextuel)
D_DATA_TABLE_INDEX	0x32	Numérique	UI_64	Optionnel. Ce TAG n'est utilisé que pour les échanges de données applicatives issues d'une zone organisée en tableau (ex. Journal météorologique ou journal des événements). Sa structure est normalisée – Voir annexe

Symbole	TAG	Format	Domaine	Description
D_DATA_TAG	0x33	String5 Ou Numérique	UI_16 + 3 x UI_8 Ou UI_16	Correspond au code TAG (UI_16) suivi des 3 indices 0, 1 et 2 (3 x UI_8) pour la donnée selon la codification d'un paramètre conformément à la SR DEV 0004 {µConfig} Ou Tag uniquement si les 3 indices sont à 0.
D_DATA_USAGE	0x34	String3	3 x UI_8	Correspond à la description physique d'un paramètre conformément à la SR DEV 0004 {µConfig} Non utilisé – Pour usage futur
D_DATA_VALUE	0x35	Variable	Variable	Valeur de la donnée
D_DATA_REQ	0x36	Numérique	2 x UI_8	Octet 1 (module 256) : Temps de récurrence en seconde (pas récurrence si 0x00) Octet 2 (division entière 256) : Nombre de récurrence. Arrêt si 0x00 et Infini si 0xFF
D_DATA_REQ_SLOT0 D_DATA_REQ_SLOT1 D_DATA_REQ_SLOT2 D_DATA_REQ_SLOT3 D_DATA_REQ_SLOT4 D_DATA_REQ_SLOT5 D_DATA_REQ_SLOT6 D_DATA_REQ_SLOT7 D_DATA_REQ_SLOT8 D_DATA_REQ_SLOT9 D_DATA_REQ_SLOT10 D_DATA_REQ_SLOT11 D_DATA_REQ_SLOT12 D_DATA_REQ_SLOT13 D_DATA_REQ_SLOT14 D_DATA_REQ_SLOT15	0x37 0x38 0x39 0x3A 0x3B 0x3C 0x3D 0x3E 0x3F 0x40 0x41 0x42 0x43 0x44 0x45 0x46	Aucune valeur associée à ce tag (longueur =0 implicitement)	s/o	Ce tag est toujours associé avec un D_DATA_VALUE dans un message IC_DATA_IN_RES (voir la conversation DATA_IN_REQ). C'est un alias qui permet d'alléger la bande passante pour ne pas répéter le D_DATA_TAG à chaque envoi récurrent

Symbole	TAG	Format	Domaine	Description
D_CIPHER_KEY	0x48	String32	32 x UI_8	Clé de chiffrement AES 256 entre l'ICom et la télécommande (est uniquement échangé durant la synchronisation)
D_SN_PERIPH	0x49	Numérique	UI_32	Numéro de série de l'appareil (MC+ ou télécommande)

Symbole	TAG	Format	Domaine	Description
D_DATA_FIRST_TABLE_INDEX	0x50	U_64 obligatoire pour l'AFSEC+	UI_64	TAG utilisé dans les conversations AF_DATA_OUT_TABLE_INDEX et AF_DATA_OUT_TABLE_INDEX pour connaître le 'table_index' du plus ancien enregistrement d'une zone organisée en tableau (ex. Journal météorologique ou journal des événements). Sa structure est normalisée – Voir annexe
D_DATA_LAST_TABLE_INDEX	0x51	U_64 obligatoire pour l'AFSEC+	UI_64	TAG utilisé dans les conversations AF_DATA_OUT_TABLE_INDEX et AF_DATA_OUT_TABLE_INDEX pour connaître le 'table_index' du dernier enregistrement (le plus récent) d'une zone organisée en tableau (ex. Journal météorologique ou journal des événements). Sa structure est normalisée – Voir annexe


Symbole	TAG	Format	Domaine	Description
D_DOWNLOAD_SECTION	0x60	UI_8 à UI_32	[1..3]	Définit la section de FLASH concernée par un téléchargement à suivre : 1 : Programme d'application 2 : Lot de tickets 3 : Catalogue de traduction
D_DOWNLOAD_NAME	0x61	String	Chaîne de caractères	Nom du fichier en téléchargement
D_DOWNLOAD_NB_RECORDS	0x62	UI_8 à UI_32	(positif)	Nombre total de S_Records (lignes) dans le fichier en téléchargement
D_DOWNLOAD_STATUS	0x63	UI_8 à UI_32	[0..9]	Indique l'état du téléchargement pour l'AFSEC+ : 0 : Téléchargement en cours et OK 1 : Téléchargement terminé OK 2 : Téléchargement refusé (section incorrecte) 3 : Téléchargement impossible (FLASH protégée) 4 : Données incorrectes 5 : Problème de flashage pendant le téléchargement
D_DOWNLOAD_RECORD	0x64	String	Ligne au format S19 (127 cars max)	Un enregistrement S_Record au format S19 de données de téléchargement
D_DOWNLOAD_END	0x65	Sans objet	Aucun	Indique la fin des données pour le téléchargement (aucun format nécessaire)

Symbole	TAG	Format	Domaine	Description
D_TEST_NB_REQS	0x71	Numérique	UI_8 à UI_32	Valeur d'un compteur de test. Théoriquement, le nombre de requêtes de TEST envoyées
D_TEST_NB_REPS	0x72	Numérique	UI_8 à UI_32	Autre valeur d'un compteur de test. Théoriquement, le nombre de réponses reçues à des requêtes de TEST

Symbole	TAG	Format	Domaine	Description
D_PACK_PAYLOAD	0xB0 PACK_IN ou PACKOUT (Peut être répété plusieurs fois dans un même message)	String de 2 octets + N octets 'raw'	Max 2 + N octets 'raw' (le nombre total d'octets est par convention exprimé par un type STRING(n))	Portion d'un data-pack pour synchronisation d'une zone de données : Octet0 = 0x12 par exemple, pour dire qu'il s'agit d'un paquet 1 sur un total de 2 Octet1 = @mot -base 0 de la zone concernée Octet[2..] = Contenu des mots de la zone à synchroniser

7.3 Table des bits pour les pictogrammes (D MENU PICTOS)

Masque de bit (hexa)	Pictogramme correspondant
0x00000001	EmB
0x00000002	EmA
0x00000004	/h
0x00000008	/min

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 44/52
	Service Développement	

Masque de bit (hexa)	Pictogramme correspondant
0x00000010	m3
0x00000020	/ (entre kg et m3)
0x00000040	kg
0x00000080	L
0x00000100	min
0x00000200	°C
0x00000400	Vb
0x00000800	Vm

7.4 Numérotation des « zones » pour les données applicatives

Numéros de zone	Identification sémantique pour l'application	Notes
0	Paramètres métrologique	
1	Paramètre superviseur	
2	Journal métrologique	Tableau d'enregistrements
3	Journal des événements	Tableau d'enregistrements
4	Données de supervision	Flux AFSEC+ => Externe uniquement
5	Ordre de gestion	Flux Externe => AFSEC+ uniquement
6	Données de date de la mémorisation à rechercher dans la carte SD	L'écriture ou la lecture dans cette zone génère des événements dans l'ICOM
7	Données de date de l'évènement à rechercher dans la carte SD	L'écriture ou la lecture dans cette zone génère des événements dans l'ICOM
8	Gestion des Index pour l'AFSEC	
9	Libre	
10	Paramètres de la carte I-Com	Les paramètres de base de la carte Com
11	Paramètres Ethernet	
12	Paramètres Wi-Fi	
13	Paramètres Bluetooth	
14	Paramètres GSM GPS	
15	Paramètres CanBUS	
16	Données GPS	
17	Paramètre et données RFID	
18	Les autres données de la carte I-Com	Accéléromètre
19	Gestion de transfert des fichiers	(A faire)

Numéros de zone	Identification sémantique pour l'application	Notes
MAX_ZONES	Numéro max d'une zone valide	

7.5 Table des codes d'erreur lors d'une transmission de donnée applicative (D_DATA_ERROR)

Code erreur	Signification
0	Pas d'erreur, transmission correcte

7.6 Structure d'un index pour les données (D_DATA_TABLE_INDEX)

Cet élément permet d'identifier un enregistrement dans une table de données.

Il est optionnel car non utilisé pour les données échangées qui ne font pas partie d'une table (dans ce cas, sa valeur implicite est 0).

Il peut avoir 2 constructions qui sont normalisées :

- Date + Heure
- Date + Numéro d'ordre dans le jour

Les différentes organisations internes sont :

Support 64 bits	Entiers longs	Tableau d'octets	Date/Heure	Date/Numéro d'ordre dans le jour	Mix Date/Heure/Chrono (*)	
U_64	UI_32[0]	UI_8[0]	Année 00-99			
		UI_8[1]	Mois 01 - 12			
		UI_8[2]	Jour 01 - 31			
	UI_8[3]	Heure 00-23	0xFF (marqueur)	Heure 00-23		
	UI_32[1]	UI_8[4]	Minute 00-59	Numéro d'ordre dans le jour (UI_32)	Minute 00-59	
		UI_8[5]	Seconde 00-59		Numéro d'ordre dans le jour (3 octets)	
		UI_8[6]	Milliseconde 000-999 ou discriminant pour unicité			
		UI_8[7]				

(*) La version 'Mix Date/Heure/Chrono' est parfois utilisée pour les comptes rendus de mesurages

7.7 Masque de saisie D MENU INPUT MASK pour IC MENU

Les conversations MENU permettent à l'ICom de demander une saisie sur le prompteur de l'AFSEC+.

Cette demande s'accompagne d'un masque de saisie pour indiquer quels caractères sont éditables dans la chaîne à éditer et pour ces caractères, quel est le domaine de caractères à proposer.

Type de saisie	Valeur du masque	Liste des caractères proposés pour ce masque
Décimale	0	0
	1	01
	2	012
	3	0123
	4	01234
	5	012345
	6	0123456
	7	01234567
	8	012345678
	9	0123456789
Hexadécimale	X	0123456789ABCDEF
Signe	+, -	+,-
Alphabétique basique	Z (majuscule)	<espace>0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
Alphabétique avec minuscule	z (minuscule)	<espace>+- 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Adresse mail	@	@ ABCDEFGHIJKLMNOPQRSTUVWXYZ.-_
Chemin d'un répertoire ou fichier	/	ABCDEFGHIJKLMNOPQRSTUVWXYZ.0123456789/- _abcdefghijklmnopqrstuvwxyz
Toutes possibilités	*	<espace>!"#\$%&'()*+,- ./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{ }~
Non éditable	#	

7.8 Index d'affichage DISP INDEX

Voir TL_ALMA_DUAL_EXI_005


(Tableau à copier dans ce document à la fin du projet)

7.9 Traces exemples d'échanges AFSEC+/ICom

Les traces ci-dessous sont issues de l'outil de simulation simulCom.py

7.9.1 Conversation DATA_OUT

```
11:31:24:331 Replying : TLVFrame: 0x02 80 00 80 03
11:31:24:503 Received : TLVFrame: 0x02 7F 0C 71 04 00 00 00 01 72 04 00 00 00 01 70 03
11:31:24:503 Replying : TLVFrame: 0x02 FF 0C 71 04 00 00 00 02 72 04 00 00 00 02 F0 03
11:31:24:659 Received : TLVFrame: 0x02 00 00 00 03
11:31:24:659 Replying : TLVFrame: 0x02 80 00 80 03
```

	SR DEV 0006 G du 21.09.2023 Protocole AFSEC+/Télécommande Alma vs iCom	Page 47/52
	Service Développement	

```

11:31:26:749 Received : TLVFrame: 0x02 03 17 31 01 00 32 08 00 00 00 00 00 01 33 85 00 01
00 00 00 35 11 01 8D 03
11:31:26:749 [{zone=0, table_index=1, tag=0001:0:0:0, value=True}]
11:31:26:749 Replying : TLVFrame: 0x02 83 00 83 03
11:31:26:905 Received : TLVFrame: 0x02 00 00 00 03
11:31:26:905 Replying : TLVFrame: 0x02 80 00 80 03
11:31:27:061 Received : TLVFrame: 0x02 00 00 00 03
11:31:27:061 Replying : TLVFrame: 0x02 80 00 80 03
11:31:27:248 Received : TLVFrame: 0x02 00 00 00 03
11:31:27:248 Replying : TLVFrame: 0x02 80 00 80 03

```

7.9.2 Conversation DATA_IN

(Ici le badge RFID (TAG=0F40:0:0:0) numéro 1234)

```

10:10:45:531 AF_INIT
10:10:45:533 Replying : TLVFrame: 0x02 81 08 01 02 00 00 02 02 00 00 8A 03
10:10:45:785 Received : TLVFrame: 0x02 00 03 07 01 03 06 03
10:10:45:787 Replying IC_DATA
10:10:45:788 Replying : TLVFrame: 0x02 84 0F 31 02 00 0A 33 85 0F 40 00 00 00 35 42 04 D2 EA 03
10:10:45:991 Received : TLVFrame: 0x02 04 00 04 03
10:10:45:992 Replying : TLVFrame: Single char code=15
10:10:47:013 Received : TLVFrame: 0x02 00 00 00 03
10:10:47:014 Replying : TLVFrame: 0x02 80 00 80 03

```

7.9.3 Conversation MENU

```

15:29:01:134 Received : TLVFrame: 0x02 01 21 01 04 00 00 00 01 03 04 00 00 75 31 07 01 02 04
42 0F C3 05 04 00 00 27 75 06 82 30 31 08 82 66 72 A1 03
15:29:01:134 AF_INIT
15:29:01:134 Replying : TLVFrame: 0x02 81 08 01 02 00 00 02 02 00 00 8A 03
15:29:01:321 Received : TLVFrame: 0x02 00 03 07 01 02 07 03
15:29:01:321 AFSEC+ Mode change : 2
15:29:01:321 Replying : TLVFrame: 0x02 80 00 80 03
15:29:01:571 Received : TLVFrame: 0x02 00 00 00 03
15:29:01:571 Replying : TLVFrame: 0x02 80 00 80 03
15:29:01:836 Received : TLVFrame: 0x02 00 00 00 03
15:29:01:836 Replying : TLVFrame: 0x02 80 00 80 03
15:29:02:086 Received : TLVFrame: 0x02 00 00 00 03
15:29:02:086 Replying : TLVFrame: 0x02 80 00 80 03
15:29:02:335 Received : TLVFrame: 0x02 00 00 00 03
15:29:02:335 Replying : TLVFrame: 0x02 80 00 80 03
15:29:02:601 Received : TLVFrame: 0x02 02 0E 10 02 00 01 07 04 00 00 00 02 08 82 66 72 80 03
15:29:02:601 received : MENU(AF)={mode_afsec:2, menu_id:1, }
15:29:02:601 replying : MENU(IC)={id:1, long_display:"top menu", on_bp_ok:2, on_bp_clear:0, }
15:29:02:601 Replying : TLVFrame: 0x02 82 1C 10 04 00 00 00 01 13 88 74 6F 70 20 6D 65 6E 75
15 04 00 00 00 02 17 04 00 00 00 00 48 03
15:29:02:772 Received : TLVFrame: 0x02 02 04 11 02 00 01 14 03
15:29:02:772 Replying : TLVFrame: Single char code=06
15:29:03:833 Received : TLVFrame: 0x02 02 0A 10 02 00 02 07 04 00 00 00 02 19 03
15:29:03:833 received : MENU(AF)={mode_afsec:2, menu_id:2, }
15:29:03:833 replying : MENU(IC)={id:2, long_display:"simple menu", on_bp_ok:100,
on_bp_menu:3, on_bp_clear:1, }
15:29:03:833 Replying : TLVFrame: 0x02 82 25 10 04 00 00 00 02 13 8B 73 69 6D 70 6C 65 20 6D
65 6E 75 15 04 00 00 00 64 16 04 00 00 00 03 17 04 00 00 00 01 62 03
15:29:04:004 Received : TLVFrame: 0x02 02 04 11 02 00 02 17 03
15:29:04:004 Replying : TLVFrame: Single char code=06
15:29:05:065 Received : TLVFrame: 0x02 02 0A 10 02 00 64 07 04 00 00 00 02 7F 03
15:29:05:065 received : MENU(AF)={mode_afsec:2, menu_id:100, }
15:29:05:065 replying : MENU(IC)={id:100, short_display:"123456", long_display:"menu showing
number", on_bp_menu:101, on_bp_clear:2, }
15:29:05:065 Replying : TLVFrame: 0x02 82 2F 10 04 00 00 00 64 12 86 31 32 33 34 35 36 13 93
6D 65 6E 75 20 73 68 6F 77 69 6E 67 20 6E 75 6D 62 65 72 16 04 00 00 00 65 17 04 00 00 00 02
DB 03
15:29:05:252 Received : TLVFrame: 0x02 02 04 11 02 00 64 71 03

```



```

15:29:05:252 Replying : TLVFrame: Single char code=06
15:29:06:313 Received : TLVFrame: 0x02 02 0A 10 02 00 65 07 04 00 00 00 02 7E 03
15:29:06:313 received : MENU(AF)={mode_afsec:2, menu_id:101, }
15:29:06:313 replying : MENU(IC)={id:101, long_display:"menu showing pictos", pictos:524287,
on_bp_menu:102, on_bp_clear:2, }
15:29:06:313 Replying : TLVFrame: 0x02 82 2D 10 04 00 00 00 65 13 93 6D 65 6E 75 20 73 68 6F
77 69 6E 67 20 70 69 63 74 6F 73 14 04 00 07 FF FF 16 04 00 00 00 66 17 04 00 00 00 02 4E 03
15:29:06:469 Received : TLVFrame: 0x02 02 04 11 02 00 65 70 03
15:29:06:469 Replying : TLVFrame: Single char code=06
15:29:07:530 Received : TLVFrame: 0x02 02 0A 10 02 00 66 07 04 00 00 00 02 7D 03
15:29:07:530 received : MENU(AF)={mode_afsec:2, menu_id:102, }
15:29:07:530 replying : MENU(IC)={id:102, short_display:"123456", long_display:"menu example",
pictos:2178, on_bp_menu:103, on_bp_clear:2, }
15:29:07:530 Replying : TLVFrame: 0x02 82 2E 10 04 00 00 00 66 12 86 31 32 33 34 35 36 13 8C
6D 65 6E 75 20 65 78 61 6D 70 6C 65 14 04 00 00 08 82 16 04 00 00 00 67 17 04 00 00 00 02 77
03
15:29:07:702 Received : TLVFrame: 0x02 02 04 11 02 00 66 73 03
15:29:07:702 Replying : TLVFrame: Single char code=06
15:29:08:762 Received : TLVFrame: 0x02 02 0A 10 02 00 67 07 04 00 00 00 02 7C 03
15:29:08:762 received : MENU(AF)={mode_afsec:2, menu_id:103, }
15:29:08:762 replying : MENU(IC)={id:103, short_display:"1234567890", long_display:"menu with
long text to see how it is handled", on_bp_menu:100, on_bp_clear:2, }
15:29:08:762 Replying : TLVFrame: 0x02 82 4C 10 04 00 00 00 67 12 8A 31 32 33 34 35 36 37 38
39 30 13 AC 6D 65 6E 75 20 77 69 74 68 20 6C 6F 6E 67 20 74 65 78 74 20 74 6F 20 73 65 65 20
68 6F 77 20 69 74 20 69 73 20 68 61 6E 64 6C 65 64 16 04 00 00 00 64 17 04 00 00 00 02 AB 03
15:29:08:950 Received : TLVFrame: 0x02 02 04 11 02 00 67 72 03
15:29:08:950 Replying : TLVFrame: Single char code=06
15:29:10:010 Received : TLVFrame: 0x02 02 0A 10 02 00 64 07 04 00 00 00 02 7F 03
15:29:10:010 received : MENU(AF)={mode_afsec:2, menu_id:100, }
15:29:10:010 replying : MENU(IC)={id:100, short_display:"123456", long_display:"menu showing
number", on_bp_menu:101, on_bp_clear:2, }
15:29:10:010 Replying : TLVFrame: 0x02 82 2F 10 04 00 00 00 64 12 86 31 32 33 34 35 36 13 93
6D 65 6E 75 20 73 68 6F 77 69 6E 67 20 6E 75 6D 62 65 72 16 04 00 00 00 65 17 04 00 00 00 02
DB 03
15:29:10:182 Received : TLVFrame: 0x02 02 04 11 02 00 64 71 03
15:29:10:182 Replying : TLVFrame: Single char code=06
15:29:11:243 Received : TLVFrame: 0x02 02 0A 10 02 00 02 07 04 00 00 00 02 19 03
15:29:11:243 received : MENU(AF)={mode_afsec:2, menu_id:2, }
15:29:11:243 replying : MENU(IC)={id:2, long_display:"simple menu", on_bp_ok:100,
on_bp_menu:3, on_bp_clear:1, }
15:29:11:243 Replying : TLVFrame: 0x02 82 25 10 04 00 00 00 02 13 8B 73 69 6D 70 6C 65 20 6D
65 6E 75 15 04 00 00 00 64 16 04 00 00 00 03 17 04 00 00 00 01 62 03
15:29:11:414 Received : TLVFrame: 0x02 02 04 11 02 00 02 17 03
15:29:11:414 Replying : TLVFrame: Single char code=06
15:29:12:475 Received : TLVFrame: 0x02 02 0A 10 02 00 03 07 04 00 00 00 02 18 03
15:29:12:475 received : MENU(AF)={mode_afsec:2, menu_id:3, }
15:29:12:475 replying : MENU(IC)={id:3, long_display:"choice menu", on_bp_ok:200,
on_bp_menu:4, on_bp_clear:1, }
15:29:12:475 Replying : TLVFrame: 0x02 82 25 10 04 00 00 00 03 13 8B 63 68 6F 69 63 65 20 6D
65 6E 75 15 04 00 00 00 C8 16 04 00 00 00 04 17 04 00 00 00 01 CD 03
15:29:12:662 Received : TLVFrame: 0x02 02 04 11 02 00 03 16 03
15:29:12:662 Replying : TLVFrame: Single char code=06
15:29:13:723 Received : TLVFrame: 0x02 02 0A 10 02 00 C8 07 04 00 00 00 02 D3 03
15:29:13:723 received : MENU(AF)={mode_afsec:2, menu_id:200, }
15:29:13:723 replying : MENU(IC)={id:200, long_display:"choice", on_bp_ok:3,
value_init:"choice 2", choice_list:"choice 1|choice 2|choice 3", }
15:29:13:723 Replying : TLVFrame: 0x02 82 3A 10 04 00 00 00 C8 13 86 63 68 6F 69 63 65 15 04
00 00 00 03 18 88 63 68 6F 69 63 65 20 32 19 9A 63 68 6F 69 63 65 20 31 7C 63 68 6F 69 63 65
20 32 7C 63 68 6F 69 63 65 20 33 F9 03
15:29:13:910 Received : TLVFrame: 0x02 02 04 11 02 00 C8 DD 03
15:29:13:910 Replying : TLVFrame: Single char code=06
15:29:14:971 Received : TLVFrame: 0x02 02 04 11 02 00 C8 DD 03
15:29:14:971 Replying : TLVFrame: Single char code=06
15:29:16:032 Received : TLVFrame: 0x02 02 14 10 02 00 03 1B 88 63 68 6F 69 63 65 20 33 07 04
00 00 00 02 8D 03
15:29:16:032 received : MENU(AF)={mode_afsec:2, menu_id:3, user_input:choice 3, }
15:29:16:032 replying : MENU(IC)={id:3, long_display:"choice menu", on_bp_ok:200,
on_bp_menu:4, on_bp_clear:1, }
15:29:16:032 Replying : TLVFrame: 0x02 82 25 10 04 00 00 00 03 13 8B 63 68 6F 69 63 65 20 6D
65 6E 75 15 04 00 00 00 C8 16 04 00 00 00 04 17 04 00 00 00 01 CD 03
15:29:16:204 Received : TLVFrame: 0x02 02 04 11 02 00 03 16 03
15:29:16:204 Replying : TLVFrame: Single char code=06
15:29:17:264 Received : TLVFrame: 0x02 02 0A 10 02 00 01 07 04 00 00 00 02 1A 03

```

```

15:29:17:264 received : MENU(AF)={mode_afsec:2, menu_id:1, }
15:29:17:264 replying : MENU(IC)={id:1, long_display:"top menu", on_bp_ok:2, on_bp_clear:0, }
15:29:17:264 Replying : TLVFrame: 0x02 82 1C 10 04 00 00 00 01 13 88 74 6F 70 20 6D 65 6E 75
15 04 00 00 00 02 17 04 00 00 00 00 48 03
15:29:17:436 Received : TLVFrame: 0x02 02 04 11 02 00 01 14 03
15:29:17:436 Replying : TLVFrame: Single char code=06
15:29:18:512 Received : TLVFrame: 0x02 02 04 11 02 00 01 14 03
15:29:18:512 Replying : TLVFrame: Single char code=06
15:29:19:573 Received : TLVFrame: 0x02 02 0A 10 02 00 02 07 04 00 00 00 02 19 03
15:29:19:573 received : MENU(AF)={mode_afsec:2, menu_id:2, }
15:29:19:573 replying : MENU(IC)={id:2, long_display:"simple menu", on_bp_ok:100,
on_bp_menu:3, on_bp_clear:1, }
15:29:19:573 Replying : TLVFrame: 0x02 82 25 10 04 00 00 00 02 13 8B 73 69 6D 70 6C 65 20 6D
65 6E 75 15 04 00 00 00 64 16 04 00 00 00 03 17 04 00 00 00 01 62 03
15:29:19:745 Received : TLVFrame: 0x02 02 04 11 02 00 02 17 03
15:29:19:745 Replying : TLVFrame: Single char code=06
15:29:20:806 Received : TLVFrame: 0x02 02 0A 10 02 00 03 07 04 00 00 00 02 18 03
15:29:20:806 received : MENU(AF)={mode_afsec:2, menu_id:3, }
15:29:20:806 replying : MENU(IC)={id:3, long_display:"choice menu", on_bp_ok:200,
on_bp_menu:4, on_bp_clear:1, }
15:29:20:806 Replying : TLVFrame: 0x02 82 25 10 04 00 00 00 03 13 8B 63 68 6F 69 63 65 20 6D
65 6E 75 15 04 00 00 00 C8 16 04 00 00 00 04 17 04 00 00 00 01 CD 03
15:29:20:993 Received : TLVFrame: 0x02 02 04 11 02 00 03 16 03
15:29:20:993 Replying : TLVFrame: Single char code=06
15:29:22:085 Received : TLVFrame: 0x02 02 04 11 02 00 03 16 03
15:29:22:085 Replying : TLVFrame: Single char code=06
15:29:22:256 Received : TLVFrame: 0x02 02 0A 10 02 00 04 07 04 00 00 00 02 1F 03
15:29:22:256 received : MENU(AF)={mode_afsec:2, menu_id:4, }
15:29:22:256 replying : MENU(IC)={id:4, long_display:"input menu", on_bp_ok:300, on_bp_menu:2,
on_bp_clear:1, }
15:29:22:256 Replying : TLVFrame: 0x02 82 24 10 04 00 00 00 04 13 8A 69 6E 70 75 74 20 6D 65
6E 75 15 04 00 00 01 2C 16 04 00 00 00 02 17 04 00 00 00 01 54 03
15:29:22:428 Received : TLVFrame: 0x02 02 04 11 02 00 04 11 03
15:29:22:428 Replying : TLVFrame: Single char code=06
15:29:23:489 Received : TLVFrame: 0x02 02 0A 10 02 01 2C 07 04 00 00 00 02 36 03
15:29:23:489 received : MENU(AF)={mode_afsec:2, menu_id:300, }
15:29:23:489 replying : MENU(IC)={id:300, long_display:"input", pictos:2178, on_bp_ok:4,
value_init:"ABCD--1234", input_mask:"ZZZZ##9999", }
15:29:23:489 Replying : TLVFrame: 0x02 82 31 10 04 00 00 01 2C 13 85 69 6E 70 75 74 14 04 00
00 08 82 15 04 00 00 00 04 18 8A 41 42 43 44 2D 2D 31 32 33 34 1A 8A 5A 5A 5A 5A 23 23 39 39
39 39 E7 03
15:29:23:660 Received : TLVFrame: 0x02 02 04 11 02 01 2C 38 03
15:29:23:660 Replying : TLVFrame: Single char code=06
15:29:24:815 Received : TLVFrame: 0x02 02 04 11 02 01 2C 38 03
15:29:24:815 Replying : TLVFrame: Single char code=06
15:29:25:969 Received : TLVFrame: 0x02 02 04 11 02 01 2C 38 03
15:29:25:969 Replying : TLVFrame: Single char code=06
15:29:27:124 Received : TLVFrame: 0x02 02 04 11 02 01 2C 38 03
15:29:27:124 Replying : TLVFrame: Single char code=06
15:29:28:278 Received : TLVFrame: 0x02 02 04 11 02 01 2C 38 03
15:29:28:278 Replying : TLVFrame: Single char code=06
15:29:29:339 Received : TLVFrame: 0x02 02 16 10 02 00 04 1B 8A 42 43 44 45 2D 2D 31 32 33 34
07 04 00 00 00 02 96 03
15:29:29:339 received : MENU(AF)={mode_afsec:2, menu_id:4, user_input:BCDE--1234, }
15:29:29:339 replying : MENU(IC)={id:4, long_display:"input menu", on_bp_ok:300, on_bp_menu:2,
on_bp_clear:1, }
15:29:29:339 Replying : TLVFrame: 0x02 82 24 10 04 00 00 00 04 13 8A 69 6E 70 75 74 20 6D 65
6E 75 15 04 00 00 01 2C 16 04 00 00 00 02 17 04 00 00 00 01 54 03
15:29:29:510 Received : TLVFrame: 0x02 02 04 11 02 00 04 11 03
15:29:29:510 Replying : TLVFrame: Single char code=06
15:29:30:587 Received : TLVFrame: 0x02 02 04 11 02 00 04 11 03
15:29:30:587 Replying : TLVFrame: Single char code=06
15:29:31:648 Received : TLVFrame: 0x02 02 0A 10 02 00 01 07 04 00 00 00 02 1A 03
15:29:31:648 received : MENU(AF)={mode_afsec:2, menu_id:1, }
15:29:31:648 replying : MENU(IC)={id:1, long_display:"top menu", on_bp_ok:2, on_bp_clear:0, }
15:29:31:648 Replying : TLVFrame: 0x02 82 1C 10 04 00 00 00 01 13 88 74 6F 70 20 6D 65 6E 75
15 04 00 00 00 02 17 04 00 00 00 00 48 03
15:29:31:835 Received : TLVFrame: 0x02 02 04 11 02 00 01 14 03
15:29:31:835 Replying : TLVFrame: Single char code=06
15:29:32:896 Received : TLVFrame: 0x02 02 0A 10 02 00 00 07 04 00 00 00 02 1B 03
15:29:32:896 received : MENU(AF)={mode_afsec:2, menu_id:0, }
15:29:32:896 MENU: Now ending the conversation (MENU ID=0)
15:29:32:896 Replying : TLVFrame: Single char code=15

```



SR DEV 0006 G du 21.09.2023

Protocole AFSEC+/Télécommande Alma vs iCom

Service Développement

Page 50/52

```
15:29:33:083 Received : TLVFrame: 0x02 00 00 00 03
15:29:33:083 Replying : TLVFrame: 0x02 80 00 80 03
15:29:33:270 Received : TLVFrame: 0x02 00 00 00 03
```

7.9.4 Conversation DOWNLOAD

```

1:05:57:00:442 Received : TLVFrame: 0x02 01 13 01 04 00 00 00 01 03 04 00 00 75 31 07 01 00 08 82
65 6E D2 03
10:57:00:442 AF_INIT
10:57:00:442 Replying : TLVFrame: 0x02 81 08 01 02 00 00 02 02 00 00 8A 03
10:57:00:598 Received : TLVFrame: 0x02 00 03 07 01 00 05 03
10:57:00:598 AFSEC+ Mode change : 0
10:57:00:598 Replying IC_DOWNLOAD for 'tickets.mot' file
10:57:00:598 Replying : TLVFrame: 0x02 86 14 60 01 02 61 8B 74 69 63 6B 65 74 73 2E 6D 6F 74 62
02 00 14 40 03
10:57:01:659 Received : TLVFrame: 0x02 06 04 63 02 00 00 63 03
10:57:01:674 Replying : TLVFrame: 0x02 86 C1 64 9D 53 30 30 43 30 30 30 30 34 32 34 35 34 41 35
35 34 39 32 45 34 43 34 46 35 34 36 37 0A 64 CF 53 33 32 35 30 30 31 30 30 30 30 30 30 30 32 33 32
31 30 30 30 30 30 30 31 30 30 30 38 30 32 62 38 31 30 30 30 34 33 36 38 36 31 37 32 37 36 36 35
37 34 30 30 30 44 35 30 32 30 31 35 37 44 30 30 38 30 30 30 30 30 30 30 30 30 30 44 32 39 44 0A 64
CF 53 33 32 35 33 30 30 31 30 30 30 32 30 30 30 44 32 44 36 30 32 32 30 30 33 30 32 30 38 32 35 33
31 33 30 32 45 33 31 33 30 37 33 30 30 44 36 30 32 30 41 30 31 44 36 30 32 32 30 30 33 30 41 30
36 32 35 33 30 33 32 36 34 32 46 30 30 34 33 0A AF 03
10:57:01:846 Received : TLVFrame: 0x02 06 04 63 02 00 00 63 03
10:57:01:861 Replying : TLVFrame: 0x02 86 F3 64 CF 53 33 32 35 30 30 31 30 30 30 30 34 30 30 42 30
32 32 35 33 30 30 33 32 36 34 32 46 30 30 30 43 30 36 32 35 33 30 33 32 36 34 30 30 30 30 44 36 30
36 32 30 41 30 31 44 36 30 32 32 30 30 33 30 44 30 36 32 35 33 30 33 33 36 34 30 30 30 30 38 35 0A
64 CF 53 33 32 35 30 30 31 30 30 30 36 30 44 36 30 32 30 41 30 31 44 36 30 32 32 30 30 33 31 34
30 36 32 35 33 30 33 32 36 34 33 41 30 30 31 35 30 36 32 35 33 30 33 32 36 34 30 30 30 30 44 36
30 32 30 41 30 31 44 36 30 32 32 30 30 33 36 39 0A 64 CF 53 33 32 35 30 30 31 30 30 30 38 30 31
36 30 36 32 35 33 30 33 32 36 34 33 41 30 30 31 37 30 36 32 35 33 30 33 32 36 34 30 30 30 30 44
36 30 32 30 41 30 35 44 36 30 32 32 30 30 33 30 30 30 36 34 39 36 45 36 34 36 35 37 38 30 30 32
31 0A CA 03
10:57:02:049 Received : TLVFrame: 0x02 06 04 63 02 00 00 63 03
10:57:02:064 Replying : TLVFrame: 0x02 86 F3 64 CF 53 33 32 35 30 30 31 30 30 30 30 41 30 44 36 30
32 30 41 30 31 44 36 30 32 32 30 30 33 31 39 30 43 32 35 33 30 33 33 37 35 32 30 36 31 37 36 36
31 36 45 37 34 30 30 30 30 30 31 42 30 38 32 30 32 35 33 30 33 38 36 43 37 35 30 30 30 30 33 46 0A
64 CF 53 33 32 35 30 30 31 30 30 30 43 30 44 36 30 32 30 41 30 31 44 36 30 32 32 30 30 33 31 41
30 43 32 35 33 30 33 33 37 35 32 30 36 31 37 30 37 32 38 41 37 33 30 30 30 30 31 43 30 38 32 30
32 35 33 30 33 38 36 43 37 35 30 30 30 30 46 37 0A 64 CF 53 33 32 35 30 30 31 30 30 30 45 30 44
36 30 32 30 41 30 32 30 30 31 32 34 35 36 45 32 30 30 36 33 36 31 37 33 32 30 36 34 36 35 32 30 36
43 36 39 37 34 36 39 36 37 36 35 32 43 30 30 44 36 30 32 30 41 30 31 30 30 31 41 36 43 36 35 36
39 0A CA 03
10:57:02:236 Received : TLVFrame: 0x02 06 04 63 02 00 00 63 03
10:57:02:251 Replying : TLVFrame: 0x02 86 F3 64 CF 53 33 32 35 30 30 31 30 30 31 30 30 37 33 32
30 37 32 38 32 37 33 37 35 36 43 37 34 36 31 37 34 37 33 32 30 36 34 36 35 32 30 36 44 36 35 37
33 37 35 37 32 36 31 36 37 36 35 30 30 44 36 30 32 30 41 30 31 30 30 32 34 36 44 38 32 44 41 0A
64 CF 53 33 32 35 30 30 31 30 30 31 32 30 36 44 36 46 37 32 36 39 37 33 38 32 37 33 32 30 37 30
36 31 37 32 32 30 36 43 32 37 36 39 36 45 36 34 36 39 36 33 36 31 37 34 36 35 37 35 37 32 32 30
36 36 36 46 36 45 37 34 32 30 36 36 36 46 38 30 0A 64 CF 53 33 32 35 30 30 31 30 30 31 34 30 36
39 30 30 44 36 30 32 30 41 30 33 44 36 30 32 32 30 30 33 31 45 30 38 32 35 33 30 33 35 36 43 37
35 32 30 34 43 30 30 43 32 30 32 30 30 31 36 44 36 30 32 30 41 30 32 44 36 30 32 32 30 30 33 38
41 0A C5 03
10:57:02:439 Received : TLVFrame: 0x02 06 04 63 02 00 00 63 03
10:57:02:454 Replying : TLVFrame: 0x02 86 F3 64 CF 53 33 32 35 30 30 31 30 30 31 36 30 33 35 30
34 32 35 33 37 37 33 30 30 43 30 30 32 30 30 30 43 44 36 30 32 30 41 30 32 44 36 30 32 32 30 30
41 43 32 30 32 30 30 31 36 44 36 30 32 32 30 30 31 30 30 30 36 34 35 35 35 35 32 32 46 42 39 0A
64 CF 53 33 32 35 30 30 31 30 30 31 38 30 30 30 30 33 34 30 34 32 35 33 32 37 33 30 30 44 34
30 32 30 30 31 30 44 35 30 32 30 31 46 42 44 30 30 38 30 30 31 32 30 30 30 30 30 30 31 45 30 30
30 41 32 38 30 36 32 35 33 35 32 45 33 35 39 31 0A 64 CF 53 33 32 35 30 30 31 30 30 31 41 30 37
33 30 30 44 34 30 32 30 30 30 34 43
```

30 34 33 30 38 32 45 32 35 33 30 33 32 36 43 37 35 30 30 30 30 44 36 30 32 32 30 30 31 33 30 0A
64 CF 53 33 32 35 30 30 31 30 30 31 45 30 30 30 30 34 34 35 35 35 35 32 30 30 44 36 30 32 30 41
30 31 33 45 30 36 32 35 33 35 36 43 37 35 30 30 30 30 33 46 30 38 32 45 32 35 33 30 33 32 36 43
37 35 30 30 30 30 44 36 30 32 32 30 30 31 43 31 0A 64 CF 53 33 32 35 30 30 31 30 30 32 30 30 30
30 30 34 34 35 35 35 35 32 30 30 44 34 30 32 30 30 30 35 43 38 30 32 30 30 32 32 44 35 30 32 30
31 41 31 44 30 30 38 30 30 31 32 30 30 30 30 30 30 33 36 30 30 30 41 30 30 30 41 34 34 35 35 43
42 0A B2 03
10:57:02:828 Received : TLVFrame: 0x02 06 04 63 02 00 00 63 03
10:57:02:828 Replying : TLVFrame: 0x02 86 42 64 B3 53 33 31 37 30 30 31 30 30 32 32 30 35 30 34
43 34 39 34 33 34 31 35 34 34 31 30 30 44 34 30 32 30 30 30 34 46 46 30 30 43 41 45 30 32 36 37
37 39 38 0A 64 8B 53 39 30 33 30 30 30 30 46 43 0A 8A 03
10:57:02:984 Received : TLVFrame: 0x02 06 04 63 02 00 00 63 03
10:57:02:984 Replying : TLVFrame: 0x02 86 03 65 11 01 F0 03
10:57:03:156 Received : TLVFrame: 0x02 06 04 63 02 00 01 62 03
10:57:03:156 DOWNLOAD OK
10:57:03:156 Replying : TLVFrame: Single char code=06
10:57:03:312 Received : TLVFrame: 0x02 00 00 00 03
10:57:03:312 Replying : TLVFrame: 0x02 80 00 80 03

7.10 Traces exemples d'échanges ICom/Télécommande

7.10.1 Conversation SYNC

A compléter

7.10.2 Conversation DATA_OUT

A compléter

7.10.3 Conversation DATA_IN

A compléter

7.10.4 Conversation DATA_IN_REQ

A compléter