

Ce document ne peut être utilisé, reproduit ou communiqué sans autorisation de la société.

n° LM-2-FSR-1-014

Ind D

Langue Fr

Visa O

Date 29/06/1999

FIPSENSOR

Manuel de l'utilisateur

Auteur

C. BATIS

Vérificateur

H. DAVID

hlp
TECHNOLOGIES

35, Rue Tournefort - 75005 Paris - France

Historique du document

Indice	Date	Page	Description
A	27/11/98	Toutes	Mise au format du document
B	27/06/1999	P13-16-26	Modifications pour boîtier type Eurocard
C	29/06/1999	P22-25	Modifications pour alimentation 24V/Paramètres de communication
D	26/10/2000	P13-16-28	Remise à jour pour la gestion des deux bits d'arrêt

Ce document ne peut être utilisé, reproduit ou communiqué sans autorisation de la société

SOMMAIRE CONTENTS

1. PRESENTATION GENERALE	5
2. ADRESSAGE PHYSIQUE, SEQUENCE DE BAPTEME	5
2.1. REPERAGE	5
2.2. ADRESSAGE	6
2.3. ARBITRE DE BUS.....	6
2.4. DESCRIPTION DES VARIABLES.....	6
2.5. VARIABLES PRECONFIGUREES.....	7
2.6. VARIABLES UTILISATEURS	9
3. CONFIGURATION OPERATIONNELLE DU FIPSENSOR, MESSAGERIE DE CONFIGURATION.....	13
3.1. CONFIGURATION OPERATIONNELLE.....	13
3.2. ACCES A LA CONFIGURATION OPERATIONNELLE.....	14
3.2.1. SYNTAXE DU MESSAGE DE CONFIGURATION	15
3.2.1.1. accès en lecture.....	15
3.2.1.2. accès en écriture	15
3.2.2. DESCRIPTION DE LA VARIABLE VECTEUR DE MESSAGERIE DE CONFIGURATION.....	15
3.3. LA LIAISON SERIE	15
3.3.1. CONFIGURATION DE LA LIAISON SERIE.....	16
3.3.2. FONCTIONNEMENT VIA LES SERVICES DE MESSAGERIE.....	16
3.3.3. FONCTIONNEMENT VIA LES SERVICES DE VARIABLES APERIODIQUES	18
3.3.4. TAMPONS DE RECEPTION.....	21
3.3.5. BROCHAGE DE LA LIAISON SERIE	21
3.4. CONFIGURATION DE LIVRAISON, FORMAT DES TRAMES	21
3.4.1. CONFIGURATION DE LIVRAISON	21
3.4.2. PARAMETRES DE COMMUNICATION WORLDIFIP	22

3.4.2.1. Paramètres de communication WorldFIP	22
3.4.2.2. Format des trames.....	22
3.5. BRANCHEMENT SUR LES CONNECTEURS, POSITIONNEMENT DES SWITCHES	24
3.5.1. FIPSENSOR VERSION CARTE	24
3.5.1.1. Connecteurs	24
3.5.1.2. Switches	24
3.5.1.3. Brochage des connecteurs et trous de fixation.....	24
3.5.2. FIPSENSOR VERSION BOITIER EUROCARD	25
3.6. EXEMPLE DE CONFIGURATION ET DE MISE EN OEUVRE	26

1. PRESENTATION GENERALE

- Equipement configurable uniquement par FIP :
 - Allocation d'identifieurs de gestion de réseau par adressage physique;
 - Allocation des identifieurs utilisateurs par mécanisme de téléchargement FIP.
 - Liaison série RS232 utilisant (selon le choix de l'utilisateur):
 - les services de messagerie périodique ou apériodique de FIP;
 - ou les services de variables apériodiques de FIP.

Cette documentation a été conçue pour expliquer pas à pas la mise en œuvre de FIPSENSORS non configurés. En premier lieu, l'utilisateur devra leur appliquer une séquence de baptême (voir Adressage physique, séquence de baptême). Puis, il configurera les paramètres FIP de son application, éventuellement les vérifiera. Enfin, il pourra configurer et utiliser son applicatif (voir : la liaison série).

Cette documentation est relative aux versions de logiciel (EPROM) 2.0.X. Ce numéro est indiqué sur l'eprom de la carte de communication.

2. ADRESSAGE PHYSIQUE, SEQUENCE DE BAPTEME

Lorsque l'on veut connecter une station FIP sur le réseau, il faut préalablement lui appliquer une séquence de baptême. Celle-ci lui précise :

- son nom de repérage, avec lequel le processus application pourra faire référence à la station indépendamment de l'adresse physique;
- son numéro de station sur le réseau, ce qu'on appelle son adresse physique ; l'adresse peut varier de 0 à 255, ce qui veut dire que l'on peut connecter physiquement jusqu'à 256 stations sur le réseau.

Le repérage d'une station par un nom permet d'avoir un niveau d'abstraction plus élevé puisqu'on travaille directement au niveau du process utilisateur ; une des entités du processus application aura localement les couples adresse physique/repère des stations sur le réseau en mémoire.

La séquence de baptême est la suite d'envoi à FIPSENSOR des variables 0x9000, *repère* et 0x9001, *adresse* par la station configuratrice. Cette séquence, parce qu'elle utilise des identifieurs universels, ne peut pas se dérouler sur le réseau du process utilisateur mais seulement sur un réseau à deux stations. Le réseau est constitué de la station produisant 0x9000/0x9001 et de la station à baptiser. En effet l'utilisation des identifieurs universels sur un réseau à plus de 2 stations connectées risquerait de configurer toutes les stations connectées aux mêmes repères et adresses.

2.1. REPERAGE

La station configuratrice envoie une fois la variable 0x9000 dont le contenu est :

- le nom de repère, chaîne de 16 octets ASCII.

La station à repérer (FIPSENSOR) acquiert son nouveau nom de repérage. (cf. Variables préconfigurées, les paramètres FIP de 0x9000).

2.2. ADRESSAGE

La station configuratrice envoie une fois la variable 0x9001 dont le contenu est :

- un octet représentant le numéro de segment FIP où la station sera connectée (de 0 à 255);
- un octet représentant l'adresse physique de la station (de 0 à 255);
- son nom de repère, chaîne de 16 octets ASCII;

La station à adresser (FIPSENSOR) vérifie si le nom de repère qu'on a précisé est le sien, et dans ce cas elle acquiert sa nouvelle adresse.
(cf. Variables préconfigurées, les paramètres FIP de 0x9001).

A la fin de l'acquisition soit du repère, soit de l'adresse, FIPSENSOR se remet en mode mesure et est prêt à recevoir soit un nouveau repère, soit une nouvelle adresse.

2.3. ARBITRE DE BUS

L'arbitre de bus proposé (avec le logiciel FIPDESIGNER) pour une séquence de baptême est le suivant :

Pour l'envoi de la variable de repérage :
séquence1 :

ID_DAT(0x9000)
SUSPEND(10)

ou

ID_DAT(0x9000)
WAIT(200ms)

Pour l'envoi de la variable d'adressage :
séquence2 :

ID_DAT(0x9001)
SUSPEND(11)

Remarque : on est obligé de temporiser 200ms pour que le FIPSENSOR sauvegarde sa nouvelle affectation de repérage ; pour l'adressage il est même conseillé de n'envoyer qu'une seule trame ID_DAT(0x9001) car après réception de 0x9001, le FIPSENSOR provoque un démarrage à froid. Il faut compter environ deux secondes pour ce redémarrage. Pour un autre logiciel que FIPDESIGNER, il suffit de passer les variables 0x9000 et 0x9001 en mode apériodique, en respectant les temporisations précédentes.

2.4. DESCRIPTION DES VARIABLES

Une fois la séquence de baptême appliquée, le FIPSENSOR dispose uniquement de variables préconfigurées. Celles-ci permettent :

- de vérifier la présence de l'équipement
- d'identifier l'équipement
- d'achever la configuration FIP de l'équipement
- de relire la configuration FIP de l'équipement
- d'appliquer une séquence de baptême à l'équipement

A l'aide de ces variables préconfigurées, l'utilisateur va à son tour configurer les paramètres FIP des variables dont il a besoin :

- pour la messagerie de configuration
- pour la communication série.

2.5. VARIABLES PRECONFIGUREES

Soit 0xXX l'adresse physique de l'interface (ce paramètre est modifiable par le réseau, cf. Adressage physique, séquence de baptême):

Voici la configuration des variables préconfigurées:

identifieurs	longueur	PDU	type	status	transfert messagerie	service	transfert varapériodique
(hexa)	(hexa)	(hexa)					
10XX	44	50	produite	pas de raf	sans	apériodique	sans
20XX	8	50	consommée	pr(500m s)	sans	apériodique	sans
21XX	7e	50	produite	pas de raf	sans	apériodique	sans
14XX	5	50	produite	pas de raf	sans	périodique	autorisé
9000	10	50	consommée	raf/pr(500m s)	sans	apériodique	sans
9001	12	50	consommée	raf/pr(500m s)	sans	apériodique	sans

Tableau 1 : Configuration FIP des variables préconfigurées dans le FipSensor. Raf. signifie Rafrâchissement requis et Pr., promptitude requise.

1) - **0x10XX** est la variable d'identification ;

Cette variable produite par FIPSENSOR permet d'identifier et d'accéder à l'état d'un équipement connecté au réseau. Elle comprend:

- des champs relatifs à son identification.
- le nom de repère de la station,

Son identifieur est 0x10XX où XX est l'adresse physique ;

elle est envoyée au démarrage de l'applicatif du FIPSENSOR, c'est-à-dire à chaque reconfiguration (FIP ou opérationnelle). Elle est bien sûr disponible pour chaque ID_DAT(0x10XX) de l'arbitre de bus.

Le contenu est une suite de 68 octets décomposés comme suit:

- 1 octet indiquant le premier champ (valeur 0x80);
 - 1 octet indiquant la longueur du premier champ (valeur 0x10);
 - le premier champ : 16 octets *constructeur*, dans lesquels sont précisés le nom du constructeur ;
- 1 octet indiquant le second champ (valeur 0x81);
 - 1 octet indiquant la longueur du second champ (valeur 0x14);
 - le second champ : 20 octets *modèle*, dans lesquels sont précisés le nom du modèle ;
- 1 octet indiquant le troisième champ (valeur 0x82);
 - 1 octet indiquant la longueur du troisième champ (valeur 0x08);
 - le troisième champ : 8 octets *numéro de version*, dans lesquels sont précisés le numéro de version logiciel ;

- 1 octet indiquant le quatrième champ (valeur 0x83) ;
 - 1 octet indiquant la longueur du quatrième champ (valeur 0x10);
 - le quatrième champ : 16 octets *nom de repère*, dans lesquels sont précisés le nom affecté à la station par la variable 0x9000.

2)- **0x20XX** est la variable de chargement de configuration FIP. Elle contient la description d'une variable. C'est avec 0x20XX que l'on va charger les paramètres FIP des variables utilisateurs. Si la description de la variable contenue dans 0x20XX est correcte, FIPSENSOR sauvegarde et va utiliser cette description. 0x20XX est composée de 8 octets, qui correspondent à la description d'une variable (voir Figure 1).

3)- **0x21XX** est la variable qui permet de relire la configuration FIP de toutes les variables configurées :

- le premier octet est le nombre de variables configurées (il possède la valeur 6 après une séquence de baptême) ;
- les octets suivants forment la concaténation des descriptions des variables configurées. Ces descriptions sont présentées par ordre croissant des identifiants.

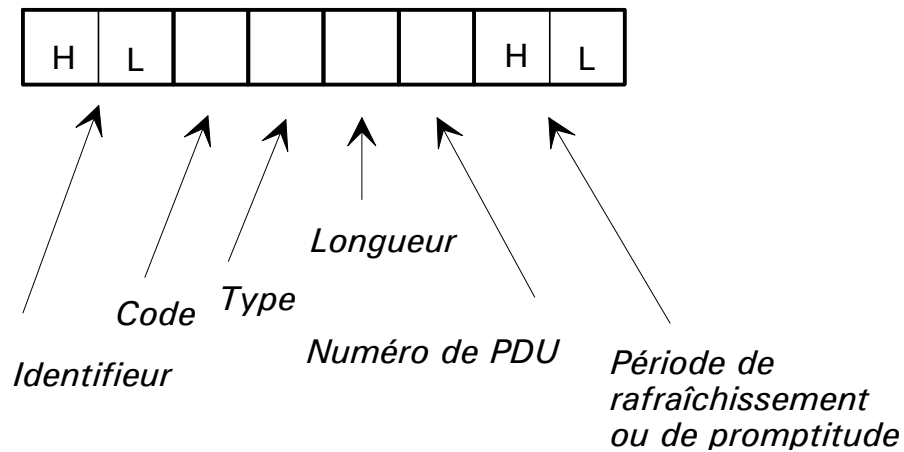


Figure 1 : description d'une variable

4)- **0x14XX** est la variable de présence.

Cette variable produite par FIPSENSOR permet pour toute entité extérieure de s'assurer de la présence de la station sur le réseau et d'accéder à l'état de l'entité de communication. Elle comprend 5 octets:

- 1 octet indiquant le premier champ (valeur 0x80);
- 1 octet indiquant la longueur du premier champ (valeur 0x03);
 - le premier champ est composé de 3 octets:
 - le premier donne la longueur de la variable d'identification (sans PDU_TYPE et PDU_LENGTH): valeur 0x44 (68 décimal);
 - le second code l'état de l'entité producteur/consommateur, doit être 0xf0;
 - le troisième code l'état de la fonction arbitre de bus, doit être 0x0.

Cette variable est produite de façon périodique, elle permet l'amorçage du trafic aperiodique de FIPSENSOR, il faut donc veiller à la présence dans l'arbitre de bus de son ID_DAT.

5)- Les identifiants **0x9000** et **0x9001** sont détaillés dans "Adressage physique, séquence de baptême".

2.6. VARIABLES UTILISATEURS

L'utilisateur, après la séquence de baptême, va devoir configurer les paramètres FIP des variables dont il a besoin. Celles-ci sont au nombre de 4 pour la version 2.0.0 :

- une variable vecteur de la messagerie de liaison série (si on utilise ce mode de communication avec la liaison série) ;
- une variable vecteur de la messagerie de configuration opérationnelle (utile si on veut lire ou changer la configuration opérationnelle du FIPSENSOR, par ex : les paramètres de communication de la liaison série) ;
- une variable de transmission série (si on utilise ce mode de communication avec la transmission de la liaison série) ;
- une variable de réception série (si on utilise ce mode de communication avec la réception de la liaison série);

Pour soulager l'applicatif de l'utilisateur désirant interfacer un FIPSENSOR, nous avons choisi, dans cette version logicielle, de n'utiliser qu'une variable de chargement (0x20XX) et qu'une variable de relecture (0x21XX) pour configurer les quatre variables utilisateur de FIPSENSOR. Pour chacune de ces variables à configurer il va falloir envoyer la description correspondante dans 0x20XX.

La description d'une variable est donnée en Figure 1.

Elle comprend:

- 1- l'identifiant (entier, 2 octets), au choix de l'utilisateur ;
- 2- le code (1 octet) associé à la variable que l'on veut configurer, voir Tableau 2
- 3- le type (1 octet) de la variable (produite/ consommée ..etc..), voir Figure 2
- 4- la longueur (1 octet) de la variable ;

5- le type de PDU (1 octet) de la variable ;

6- période du status temporel rafraîchissement ou promptitude (2 octets) en ms, ex: 0x1f4 équivaut à 500ms.

Cette période n'est significative que lorsque le bit b1 de "type" est à 1 et b6 à 0, ou bien lorsque le bit b2 de "type" est à 1 et b6 à 1 (voir

Figure 2 : octet "type" de la description d'une variable).

Si la configuration trouvée dans 0x20XX est cohérente, FIPSENSOR sauvegarde la configuration FIP de la variable, et modifie en conséquence le contenu de la variable 0x21XX. 0x21XX fait alors l'objet d'une demande de transfert aperiodique sur le réseau.

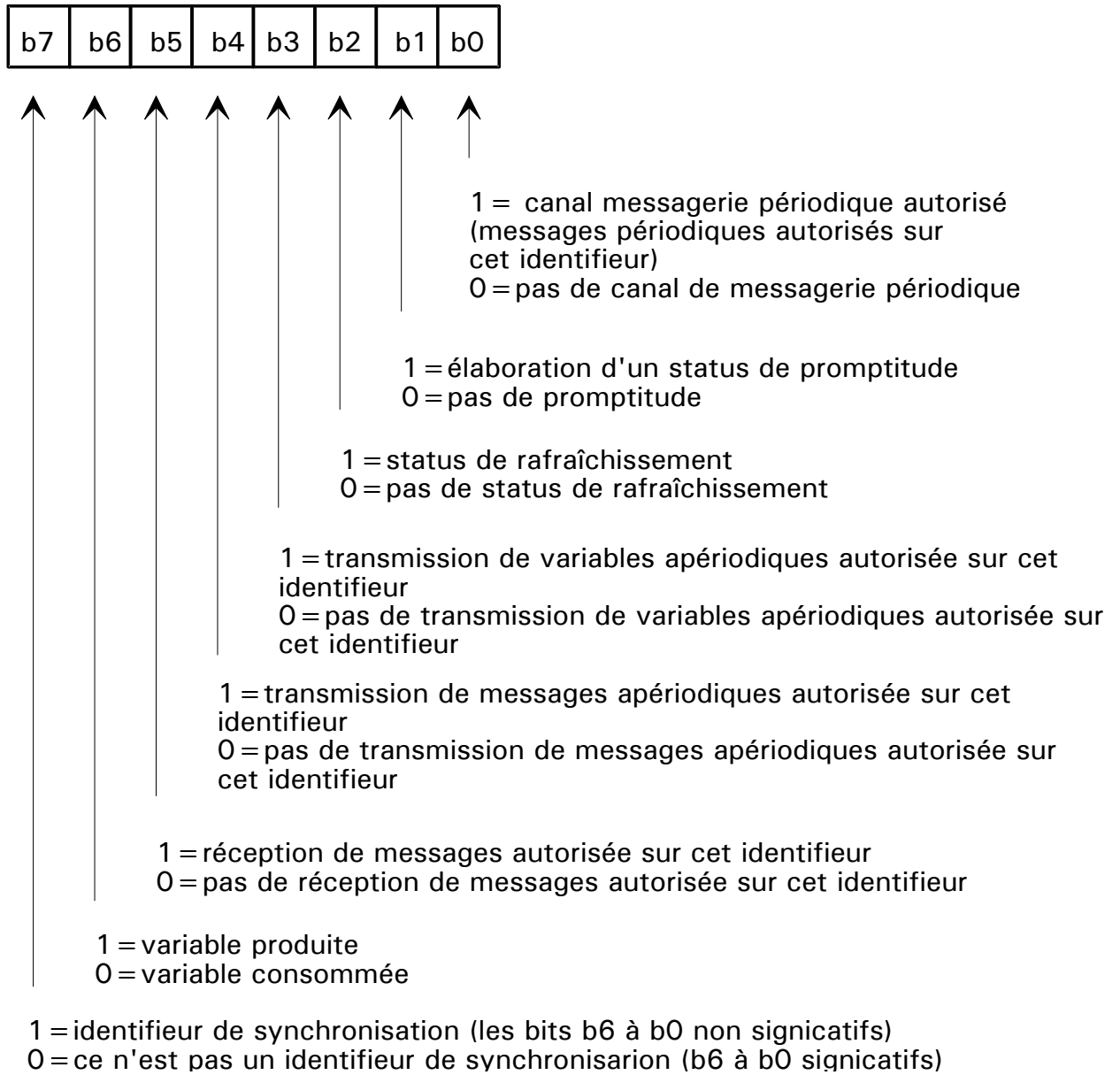
Si la configuration trouvée dans 0x20XX est incohérente, FIPSENSOR ne modifie en aucune manière 0x20XX et ne sauvegarde aucun paramètre de la variable décrite; en conséquence 0x21XX reste inchangée.

Pour enlever une variable utilisateur, une seule méthode: il faut reprendre la configuration à son début et appliquer une séquence de baptême. Toute séquence de baptême appliquée efface les variables utilisateurs.

n° d'identifieur	fonction associée	code de la fonction	Type
0x10XX	Identification	0x0	Apériodique
0x20XX	Chargement description d'une variable	0x1	Apériodique
0x21XX	Lecture des descriptions des variables	0x2	Apériodique
0x14XX	Présence	0x3	Périodique
0x9000	Repère	0x4	Apériodique
0x9001	Adresse	0x5	Apériodique
suivant la description envoyée	Variable vecteur de la messagerie de liaison série	0x6	Périodique
suivant la description envoyée	Variable vecteur de la messagerie de configuration	0x7	Périodique
suivant la description envoyée	Variable réception de la liaison série	0x8	Apériodique
suivant la description envoyée	Variable transmission de la liaison série	0x9	Apériodique
suivant la description envoyée	Synchronisation	0xa	Synchro

Tableau 2 : codes associés aux variables

Remarque: la variable de synchronisation n'est pas utilisée dans les versions logicielles 2.0.X .



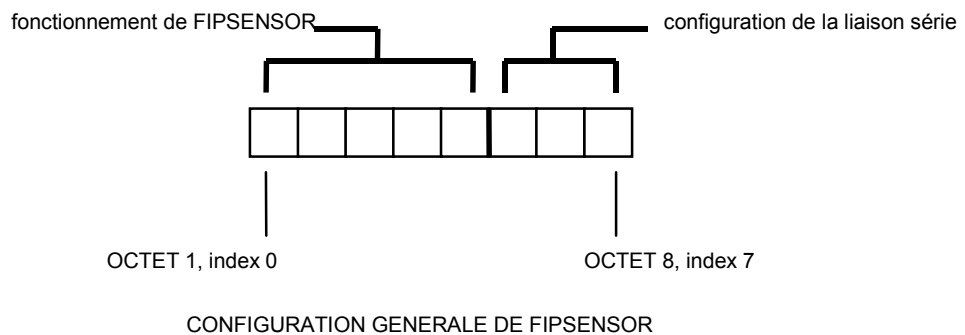
Ce document ne peut être utilisé, reproduit ou communiqué sans autorisation de la société

Figure 2 : octet "type" de la description d'une variable

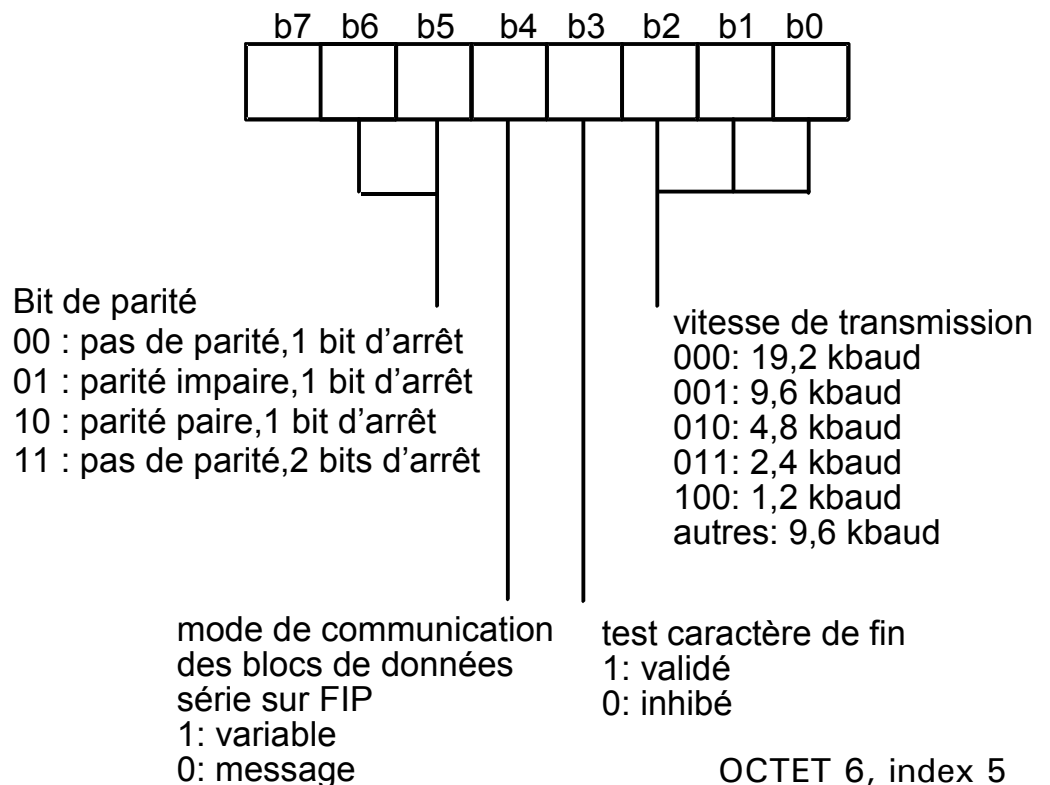
3. CONFIGURATION OPERATIONNELLE DU FIPSENSOR, MESSAGERIE DE CONFIGURATION

3.1. CONFIGURATION OPERATIONNELLE

La configuration opérationnelle se compose de huit octets. Ils permettent d'accéder au fonctionnement des différentes interfaces de FIPSENSOR. Dans la version de logiciel 2.0.0, la seule interface admise est la liaison série, en conséquence les 5 premiers octets de la configuration sont inutilisés et les trois octets restants sont relatifs au fonctionnement de la liaison série.



Les trois derniers octets liés à l'utilisation de la RS232 sont décrits ci-dessous :



- Octet 7, index 6 : caractère de fin délimitant l'envoi d'un message ou d'une variable sur FIP contenant les octets reçus de la liaison série (i.e. caractère délimitant un bloc de données série).

- Octet 8, index 7 : nombre maximum d'octets reçus délimitant l'envoi d'un message sur FIP contenant les octets reçus de la liaison série (i.e. longueur max. d'un bloc de données série).

3.2. ACCES A LA CONFIGURATION OPERATIONNELLE

L'accès à la configuration opérationnelle de FIPSENSOR est assurée par la messagerie FIP.

Si une station configuratrice veut lire ou modifier les 8 octets de configuration opérationnelle, elle envoie un "message de configuration". A la réception de celui-ci, FIPSENSOR acquitte par l'envoi du "message d'acquiescement" contenant les 8 octets de la configuration opérationnelle. Le message d'acquiescement possède toujours comme identifiant destinataire, l'identifiant source du message de configuration ; ainsi c'est toujours à la station configuratrice que le message d'acquiescement est envoyé (voir Figure 3).

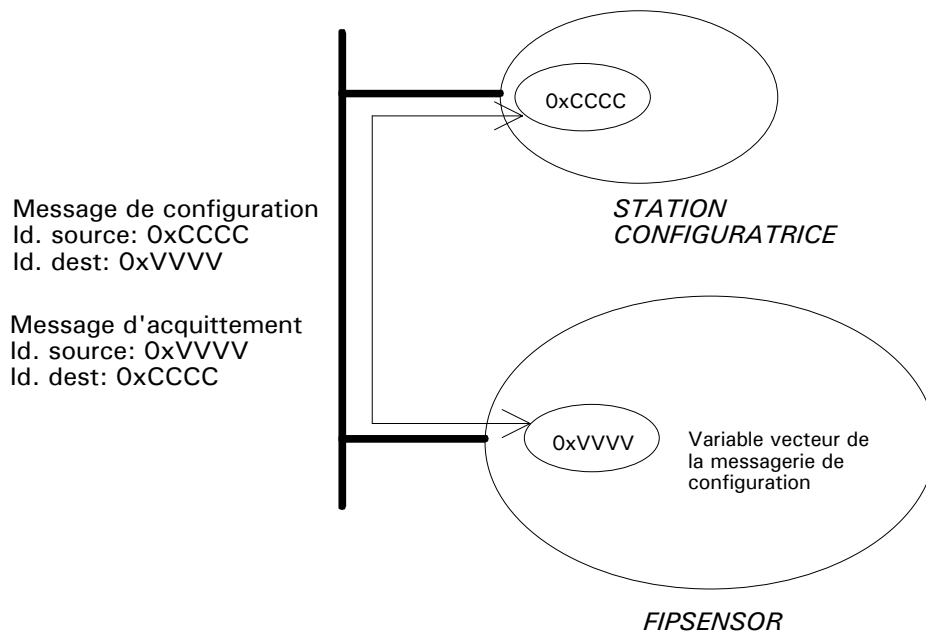


Figure 3 : messagerie de configuration

3.2.1. SYNTAXE DU MESSAGE DE CONFIGURATION

3.2.1.1. accès en lecture

Il suffit de mettre les deux caractères ASCII 'rd' (read device) dans le message de configuration de la station configuratrice. Le message d'acquittement que cette station recevra, contiendra les huit octets de la configuration opérationnelle.

Exemple :

message de configuration 0xCCCC->0xVVVV ('rd') message émis,
message reçu 0xVVVV->0xCCCC (0x00 0x00 0x00 0x00 0x00 0x01
0x0d 0x80),

3.2.1.2. accès en écriture

Il suffit de mettre, dans le message de configuration, les deux caractères ASCII 'wd' (write device) suivi de l'index pointant sur la zone à écrire (index de 0 à 7), puis le nombre d'octet à écrire suivi de ces octets ;

Pour l'octet n°6 ceci donne : 'wd' 5 1 XX

5 est l'index (l'index démarre de 0), le nombre d'octet à écrire (1), suivi de l'octet XX à écrire.

3.2.2. DESCRIPTION DE LA VARIABLE VECTEUR DE MESSAGERIE DE CONFIGURATION

La variable vecteur de messagerie 0xVVVV doit être correctement configurée. Lors de l'envoi de la description de la variable en phase de configuration FIP, il faut donc veiller aux détails suivants :

1- l'octet "code" doit être égal à 0x7; le bit b7 de "type" est à 0 ;

2- pour autoriser la réception de messages sur cet identifieur, le bit b5 de "type" doit être mis à 1 ;

3- il faut autoriser la transmission de message.

Pour cela deux cas de figure se présentent :

- la messagerie périodique:

les bits suivants de "type" sont positionnés comme suit,

b0=1,

b4=0,

d'autre part l'arbitre de bus doit contenir ID_MSG(0xVVVV).

- la messagerie aperiodique :

les bits suivants de "type" sont positionnés comme suit,

b0=0,

b4=1,

b6=1 (variable produite) et l'arbitre de bus doit contenir ID_DAT(0xVVVV) pour amorcer la requête aperiodique de FIPSENSOR.

4- Les autres paramètres (longueur, PDU, identifieur, rafraîchissement, promptitude) sont au choix de l'utilisateur.

3.3. LA LIAISON SERIE

Un ensemble de données *transmis* sur la liaison série par FIPSENSOR provient du contenu d'un message reçu de FIP ou bien d'une variable reçue de FIP. Ce choix entre variables et messages est effectué lors de la configuration du FIPSENSOR, et c'est un choix exclusif.

Dans le cas messagerie, on appellera ce message FIP, "message de transmission série". Dans le cas variable, on appellera cette variable FIP, "variable de transmission série".

Un ensemble de données *reçu* de la liaison série par FIPSENSOR est transmis dans le contenu d'un message FIP ou bien dans une variable produite de FIP. Ce choix entre variables et messages est effectué lors de la configuration du FIPSENSOR, et c'est un choix exclusif.

Dans le cas messagerie, on appellera ce message FIP, "message de réception série". Dans le cas variable, on appellera cette variable FIP, "variable de réception série".

Trois octets de la configuration opérationnelle permettent de gérer le mode de fonctionnement de cet interface de FIPSENSOR.

3.3.1. CONFIGURATION DE LA LIAISON SERIE

La transmission et la réception s'effectuent avec la configuration suivante:

- 8 bits de données;
- 1 bit de stop;
- Selon le choix : pas de bit de parité et 1 bit d'arrêt ou 1 bit de parité paire et 1 bit d'arrêt ou 1 bit de parité impaire et 1 bit d'arrêt ou pas de bit de parité et 2 bit d'arrêts.
- les vitesses configurables sont 1200,2400,4800,9600,19200 bauds.

Trois octets de la configuration opérationnelle permettent de gérer le mode de fonctionnement de cet interface de FIPSENSOR. Ceux-ci sont les trois derniers octets de la configuration et ils sont décrits dans le paragraphe "messagerie de configuration".

- L'octet 6 de cette configuration permet :
 - 1- de configurer la vitesse entre 1200 et 19200 bauds (bits b0 à b2),
 - 2- de déterminer si on utilise un caractère de fin (caractère de fin donné par l'octet 7) pour délimiter les blocs de données en réception sur la liaison série (bit b3),
 - 3- de choisir le mode de communication FIP, soit par variables, soit par messages (bit b4).
 - 4- de déterminer si l'on désire avoir un bit de parité paire ou impaire, ou pas de bit de parité, et si l'on désire 1 ou 2 bits d'arrêt.
- L'octet 7 est le caractère de fin (valeur entièrement configurable par la messagerie de configuration).
- L'octet 8 est la longueur d'un bloc de données reçu devant être envoyé sur FIP.

3.3.2. FONCTIONNEMENT VIA LES SERVICES DE MESSAGERIE

Si le bit b4 de l'octet 6 de la configuration opérationnelle est positionné sur 0, on utilise la messagerie. Un message FIP est défini par son identificateur source et son identificateur destinataire.

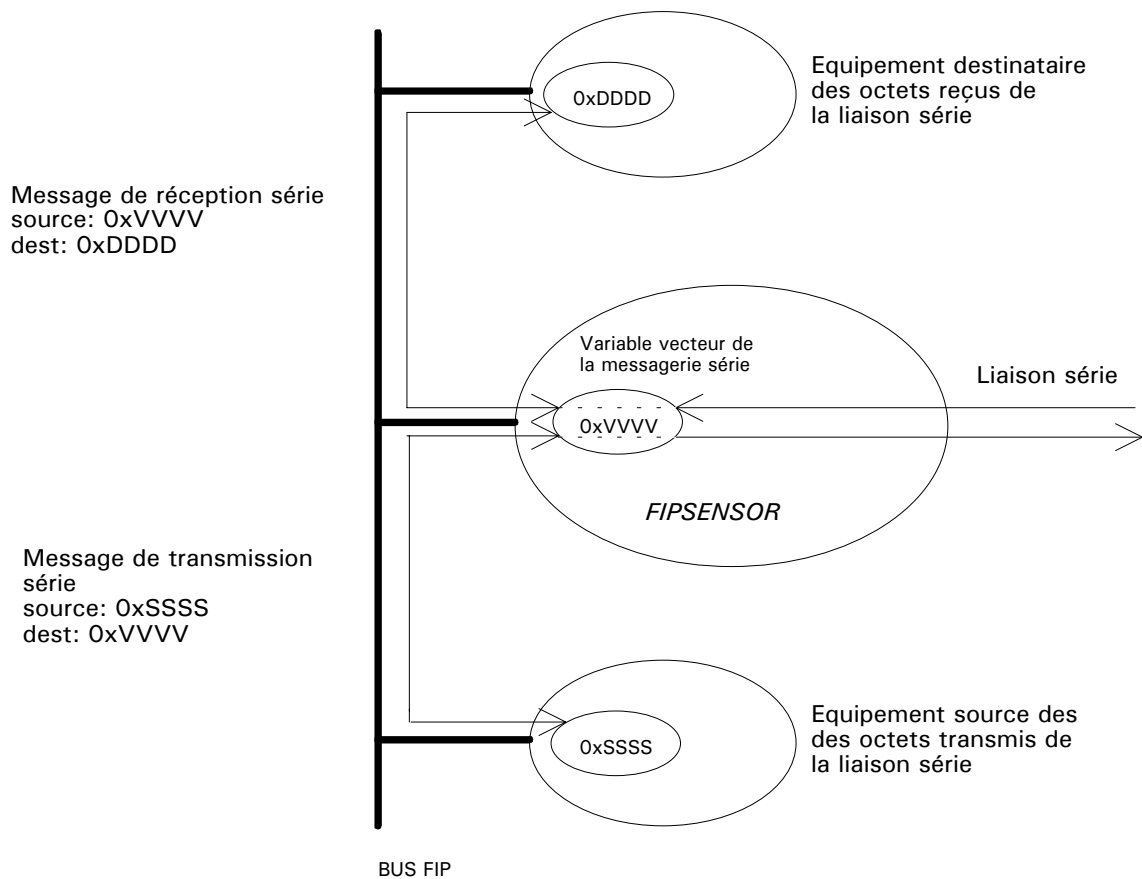


Figure 4 : messages FIP échangés lors d'une communication via la liaison série

la Figure 4 présente le fonctionnement en mode message.

En réception, tout bloc de données délimité par le caractère de fin (b3 de l'octet 6 de la configuration opérationnelle à 1) ou délimité par la longueur maximale (caractère de fin non rencontré ou bien non validé, b3 à 0) est transmis dans le message de réception série, caractère de fin compris (si il est présent). En transmission, tout caractère contenu dans le message de transmission série est retransmis sur la liaison série. Ces messages peuvent avoir une longueur de 250 octets au maximum, en conséquence l'octet 8 de la configuration opérationnelle est au maximum 250 (0xfa).

La variable vecteur de messagerie 0xVVVV doit être correctement configurée. Lors de l'envoi de la description de la variable en phase de configuration FIP, il faut donc veiller aux détails suivants :

- 1- l'octet "code" doit être égal à 0x6; le bit b7 de "type" est à 0 ;
- 2- pour autoriser la réception de messages sur cet identifieur le bit b5 de "type" doit être mis à 1 ;
- 3- il faut autoriser la transmission de message, deux cas de figure se présentent :

- la messagerie périodique :
les bits suivants de "type" sont positionnés comme suit,
b0=1,
b4=0,
d'autre part l'arbitre de bus doit contenir ID_MSG(0xVVVV).

- la messagerie aperiodique:

les bits suivants de "type" sont positionnés comme suit,

b0=0,

b4=1,

b6=1 (variable produite) et l'arbitre de bus doit contenir ID_DAT(0xVVVV) pour amorcer la requête aperiodique de FIPSENSOR.

4- Les autres paramètres (longueur, PDU, identifieur, rafraîchissement, promptitude) sont au choix de l'utilisateur.

D'autre part FIPSENSOR doit connaître l'identifieur destinataire du message de réception série.

Celui-ci est configuré via la messagerie de configuration : il faut mettre dans ce message les deux caractères ASCII "wf" suivi du numéro d'identifieur 0xDDDD (partie haute puis partie basse).

Lorsque FIPSENSOR reçoit ce message, il réémet toujours via la messagerie de configuration (message d'acquittement) deux octets correspondant au numéro de l'identifieur, destinataire effectivement configuré.

On peut lire l'identifieur destinataire du message de réception série déjà configuré. Il suffit de mettre dans le message de configuration les deux caractères ASCII "rf" ; lorsque FIPSENSOR reçoit ce message, il réémet toujours via la messagerie de configuration (message d'acquittement) deux octets correspondant au numéro de l'identifieur destinataire effectivement configuré.

3.3.3. FONCTIONNEMENT VIA LES SERVICES DE VARIABLES APERIODIQUES

Si le bit b4 de l'octet 6 de la configuration opérationnelle est positionné à 1, on utilise deux variables aperiodiques (voir

Figure 5 : variables FIP échangées lors d'une communication via la liaison série).

En réception, tout bloc de données délimité par le caractère de fin (b3 de l'octet 6 de la configuration opérationnelle à 1) ou délimité par la longueur maximale (caractère de fin non rencontré ou bien non validé, b3 à 0) est transmis dans la variable de réception série, caractère de fin compris (si il est présent).

Si la longueur du bloc de données est inférieure à la longueur de la variable, le reste des octets de la variable est complété par des 0.

Attention, la longueur d'un bloc de donnée (octet 8 de la configuration opérationnelle) ne peut dépasser la longueur de la variable de réception.

En transmission, tout caractère contenu dans la variable de transmission série est retransmis sur la liaison série jusqu'au caractère de fin ou la longueur maximale d'un bloc de donnée.

Lors de l'envoi des descriptions des variables en phase de configuration FIP, il faut veiller aux détails suivants :

- pour la variable de réception série 0xRRRR :

1- l'octet "code" doit être égal à 0x8 ; le bit b7 de "type" est à 0 ;

2- cette variable est produite, le bit b6 de "type" est à 1 ;

3- cette variable est transmise en aperiodique sur FIP, il faut donc une variable initiatrice de la requête aperiodique. FIPSENSOR impose pour celle-ci la variable de présence (periodique) et la transmission de variable aperiodique est autorisée sur 0x14XX. Il s'en suit qu'il faut juste vérifier la présence de ID_DAT(0x14XX) dans l'arbitre de bus.

- pour la variable de transmission série 0xTTTT :

1- l'octet "code" doit être égal à 0x9; le bit b7 de "type" est à 0 ;

2- cette variable est consommée, le bit b6 de "type" est à 0.

- Pour ces deux variables, l'octet de longueur dans la description ne peut dépasser la valeur :

- 125 (0x7d) octets si un status de rafraîchissement est requis ;
- 126 (0x7e) octets si il n'y a pas de status de rafraîchissement requis ;

en conséquence l'octet 8 de la configuration opérationnelle (qui correspond à la longueur d'un bloc de données reçu devant être envoyé sur FIP) doit être positionné au maximum à 125 ou 126 et de toute façon doit être inférieure ou égale à la longueur des variables de réception 0xRRRR ou de transmission 0xTTTT.

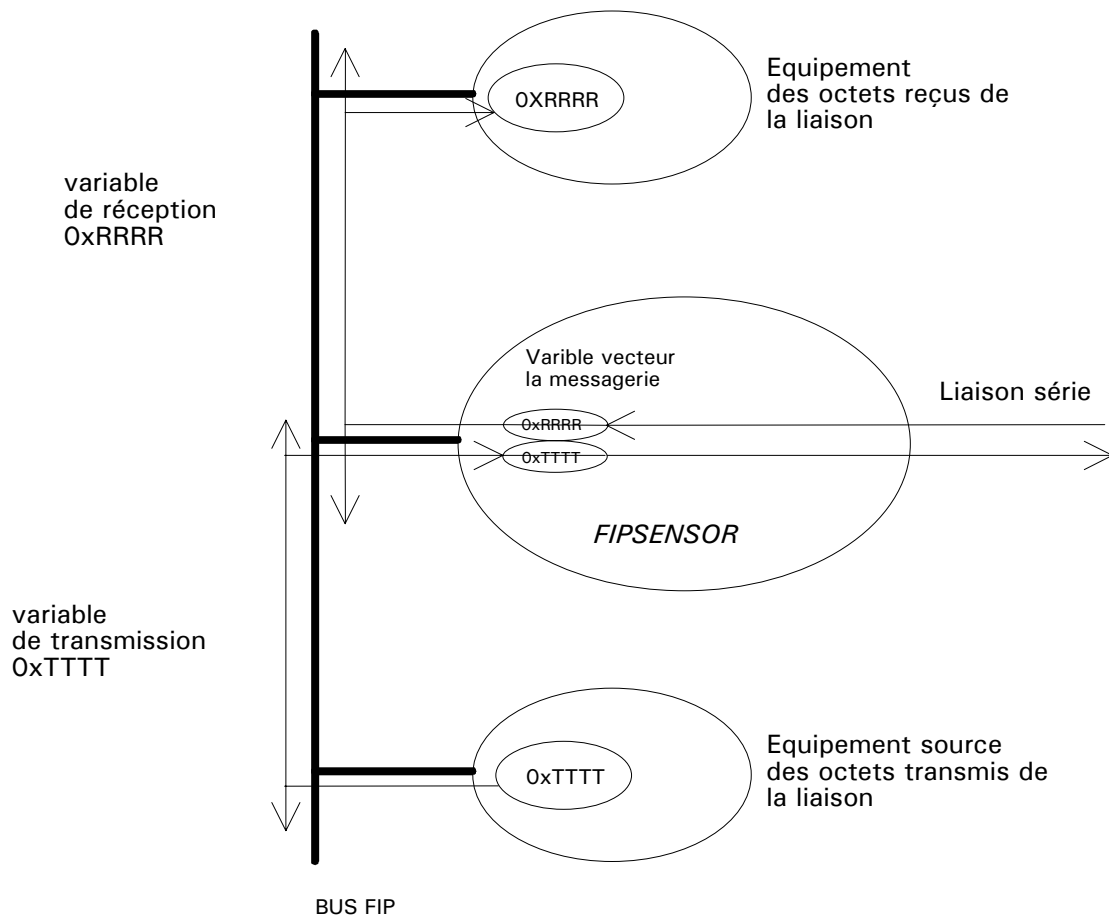


Figure 5 : variables FIP échangées lors d'une communication via la liaison série

3.3.4. TAMPONS DE RECEPTION

Pour adapter les flux de données de part et d'autre de l'interface FIP/RS232, la version logicielle 2.0.X propose deux tampons d'une profondeur de 8 blocs de données.

La règle de part et d'autre de l'interface FIP/RS232 est la suivante : un message n'est accepté chez le récepteur que s'il peut être stocké. En aucun cas un message ne peut écraser un message reçu antérieurement.

Ainsi :

- En réception de la liaison série : FIPSENSOR dispose d'un tampon de 8 blocs de données. Tant que ce tampon n'est pas plein, tout bloc de données reçu de la liaison série dans ce tampon est envoyé sur FIP, tandis que d'autres blocs peuvent être reçus.

Si le tampon est plein (8 blocs à transmettre sur FIP) les nouveaux blocs de données reçus de la liaison série seront perdus, tandis que les 8 blocs du tampon seront transmis sur FIP.

- De même en réception de FIP : FIPSENSOR dispose d'un tampon de 8 variables ou messages FIP (suivant le mode choisi). Tant que ce tampon n'est pas plein, tout message ou variable reçu de FIP dans ce tampon est envoyé sur la liaison série, tandis que d'autres variables ou messages peuvent être reçus.

Si le tampon est plein (8 blocs de données à transmettre sur la liaison série) les nouveaux messages ou variables reçus de FIP seront perdus, tandis que les 8 du tampon seront transmis sur la liaison série.

3.3.5. BROCHAGE DE LA LIAISON SERIE

La RS232 se connecte via un connecteur de type mini-DIN sur la version boîtier Eurocard. Le brochage est indiqué sur la Figure 6, et comporte 3 fils, tx pour la transmission, rx pour la réception et la masse.

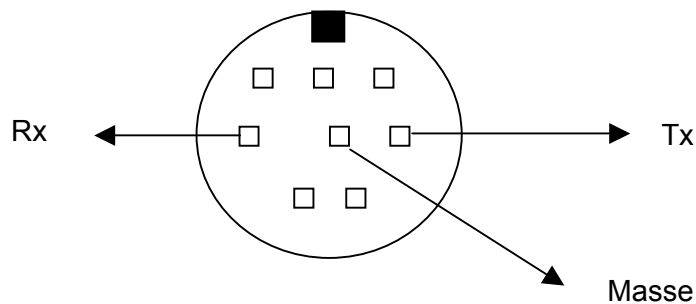


Figure 6 : connecteur mini-DIN vu de la face avant de FipSensor.

3.4. CONFIGURATION DE LIVRAISON, FORMAT DES TRAMES

3.4.1. CONFIGURATION DE LIVRAISON

Le FIPSENSOR est livré prêt à fonctionner avec une configuration de livraison. Celle-ci est la suivante :

- adresse physique = 0x59 ;
- nom de repère = numéro de série ;
- variable vecteur de messagerie de liaison série préconfigurée avec les paramètres suivants :
 - identifieur = 0x3000
 - produite
 - transmission et réception de message autorisés
 - longueur = 1 octet
 - PDU = 0x40 ;
- variable vecteur de messagerie de configuration préconfigurée avec les paramètres suivants :
 - identifieur = 0x6000
 - produite
 - transmission et réception de message autorisés
 - longueur = 1 octet
 - PDU = 0x40 ;
- variable de réception de liaison série préconfigurée avec les paramètres suivants :
 - identifieur = 0x4000
 - produite
 - longueur = 126 octets
 - PDU = 0x40 ;
- variable de transmission de liaison série préconfigurée avec les paramètres suivants :
 - identifieur = 0x5000
 - consommée
 - longueur = 126 octets
 - PDU = 0x40 ;
- identifieur configuré destinataire du message de réception série est 0x559 ;
- la liaison série configurée à 9600 bauds, caractère de fin validé et égal à "cr" (0xd), longueur de bloc 126 octets, mode variable aperiodique.

L'exemple de mise en œuvre dans le dernier paragraphe de ce manuel est la programmation de cette configuration de livraison.

3.4.2. PARAMETRES DE COMMUNICATION WORLDFIP

3.4.2.1. Paramètres de communication WorldFIP

Le FipSensor est un matériel architecturé autour des composants de communication de CEGELEC c'est-à-dire FullFIP2 et FielDrive¹. A 1Mb/s, le matériel communique dans le mode type SLOWFIP de CEGELEC, à savoir, temps de retournement 41µs, et temps de silence, 290µs². Pour tout autre valeur, veuillez nous consulter.

3.4.2.2. Format des trames

Depuis la publication de la norme CENELEC EN61158-2, le format des pré- et post-ambules des trames FIP au niveau couche physique a changé (voir Figure 6 et Figure 7). Pour rester compatible avec les deux types de trame, deuxeproms sont livrées avec un FIPSENSOR. Le sens de l'insertion du composant est donné par le détrompage du support et/ou de la sérigraphie.

Sur les deux figures ci-dessous les trames FIP sont constituées de trois parties :

¹ FullFIP2 et FielDrive sont des marques déposés de CEGELEC

² Pour les utilisateurs de FipDesigner : ces valeurs de temporisations ne peuvent qu'être approchées. Il s'en suit que dans les configurations de FipDesigner on utilise les valeurs suivantes, 41,25µs et 264µs.

- 1- La séquence de début de trame;
- 2- Le champ contrôle et données;
- 3- La séquence de fin de trame.

La couche physique ajoute pour sa part la séquence de début et de fin de trame à toute trame émise.

La séquence de début de trame se décompose comme suit :

- préambule (PRE), qui permet aux récepteurs de se synchroniser sur l'horloge de l'émetteur;
- délimiteur de début de trame (DDT), qui indique à la couche liaison de données où elle peut trouver le début de l'information utile, c'est-à-dire le champ de contrôle et de données (CED).

La séquence de fin de trame est constituée plus simplement :

- délimiteur de fin de trame (DFT), qui indique à la couche liaison de données la fin de l'information utile (CED).

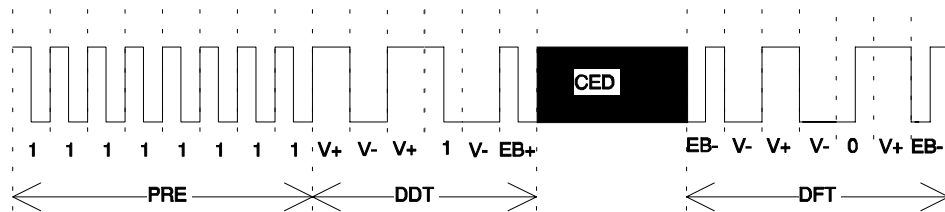


Figure 7 : Séquences ajoutées par la couche physique à une trame FIP. Ces séquences sont celles spécifiées par la norme NF C 46604. Elles sont pratiquement appelées trames FIP.

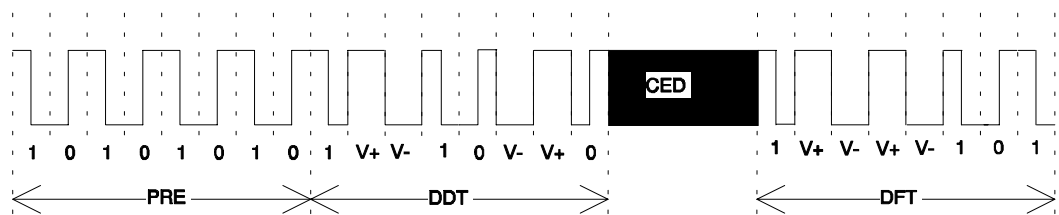


Figure 8 : Séquences ajoutées à une trame FIP par la couche physique. Celles-ci sont spécifiées par la norme CENELEC EN61158-2. Elles sont pratiquement appelées trames WorldFIP.

L'indication sous le numéro de version de l'eprom permet de reconnaître l'eprom nécessaire à l'application utilisateur :

- eprom « **fip** » assure des trames du type de la **Figure 6**;
- eprom « **wfip** » assure des trames du type de la **Figure 7**

3.5. BRANCHEMENT SUR LES CONNECTEURS, POSITIONNEMENT DES SWITCHES

3.5.1. FIPSENSOR VERSION CARTE

Les paragraphes suivants ne se rapportent qu'aux FIPSENSOR livrés sans boîtier (pour insertion dans un système déjà existant). Le connecteur 50 broches permet la connexion au bus du processeur 80C31. Pour le brochage nous consulter.

3.5.1.1. Connecteurs

La carte "FF2-8031 V0.2" dispose de trois connecteurs :

1- (optionnel) l'alimentation 24V DC, sur un connecteur orange à deux pôles: masse pôle "ALIM-" et 24V pôle "ALIM+". Elle est protégée contre l'inversion de pôle. FIPSENSOR consomme 60 mA sous 24V.

2- (optionnel) la liaison série, sur un connecteur orange à trois pôles : masse pôle "GND", transmission pôle "TXD", réception pôle "RXD".

3- le connecteur FIP, de couleur vert à deux pôles: le branchement D+ D- est indiqué en sérigraphie : D+ = D; D- = Dn. Attention l'inversion D+ D- rend impossible toute communication par FIP.

3.5.1.2. Switches

La position des switches est indiquée en grisé sur la Figure 9:

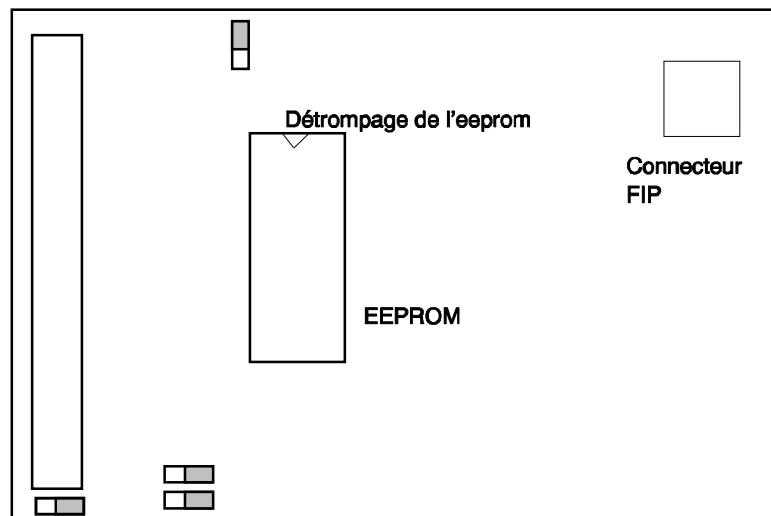


Figure 9 : positionnement des switches

3.5.1.3. Brochage des connecteurs et trous de fixation

Le brochage des connecteurs explicité ci-dessus et la position des trous de fixation sont indiqués en Figure 10 :

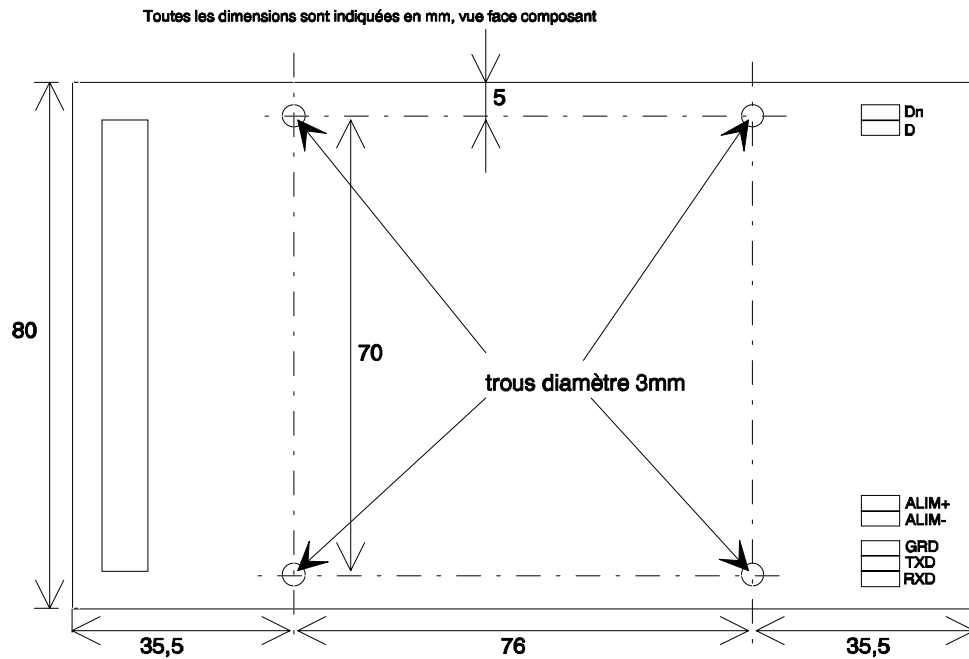


Figure 10 : position des trous de fixation

3.5.2. FIPSENSOR VERSION BOITIER EUROCARD

Cette version dispose sur sa face avant les éléments suivants :

- Leds COM et ERR : valide le fonctionnement du processeur de communication. La Led COM allumée indique l'utilisation du processeur de communication. Attention : Elle n'indique pas la communication effective sur le réseau. La Led ERR indique une erreur sur le processeur de communication. Dans ce cas, un redémarrage à froid est aussitôt (2 secondes environ) réalisé par le superviseur d'activité embarqué dans cette version.
- Bouton RESET : permet le redémarrage à froid du FipSensor.
- Bouton Factory setup : redémarrage à froid et rappel de la configuration de livraison. Non utilisé dans la version logicielle 2.0.X.
- Connecteur mini-DIN RS232/12C : connecteur de liaison série. Pour la version logicielle 2.0.X, il s'agit d'une RS232 trois fils. Le brochage est donné en paragraphe 3.3.5.
- Connecteur FIP : DB9 mâle Fip normalisé. Broche 6, D+ et broche 7, D-.

L'alimentation est réalisée en face arrière par un bornier à vis. L'alimentation est 24v continu/250mA. Elle est protégée contre les inversions, le niveau d'entrée doit être compris entre 21,6 et 26,4v. Le bornier dispose d'une indication écrite pour distinguer la borne positive et la borne négative. Cependant, si cette indication disparaît, la borne positive est celle de droite et la négative celle de gauche, en positionnant le bornier comme suit : Bornier face à vous, chambres serre-fils visibles, vis de serrage sur le dessus.

3.6. EXEMPLE DE CONFIGURATION ET DE MISE EN OEUVRE

1) Séquence de baptême:

positionnement à l'adresse physique n°0x59, avec comme nom de repère "n951101-1-5"

a) envoi de la variable 0x9000 avec comme contenu:

0x9000 ("n951101-1-5 ") en ASCII ou bien

0x9000 (0x6e 0x39 0x35 0x31 0x31 0x30 0x31 0x2d 0x31 0x2d 0x35 0x20 0x20 0x20 0x20 0x20) en hexadécimal.

b) envoi de la variable 0x9001 avec comme contenu:

0x9001 (0x0 0x59 "n951101-1-5 ") ou bien

0x9001 (0x0 0x59 0x6e 0x39 0x35 0x31 0x31 0x30 0x31 0x2d 0x31 0x2d 0x35 0x20 0x20 0x20 0x20 0x20) en hexadécimal.

La séquence de baptême est achevée, FIPSENSOR doit envoyer périodiquement la variable de présence 0x1459. Si celle-ci n'est pas reçue, la séquence de baptême doit être reprise à son début. Après la séquence de baptême, les configurations FIP des variables utilisateurs sont effacées, la variable 0x2159 affiche le contenu suivant :

6	10	59	0	40	44	50	0	0	14	
59	3	48	5	50	0	0	20	59	1	2
8	50	1	f4	21	59	2	40	7e	5	
0	0	0	90	0	4	6	10	50	1	f4
90	1	5	6	12	50	1	f4	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

2) On va maintenant envoyer la configuration FIP de la variable vecteur de messagerie de liaison série, nous allons prendre comme identifieur pour cette variable 0x3000 :

0x2059 (0x30 0x00 0x6 0x70 0x1 0x40 0x0 0x0)

Si la description de la variable vecteur de messagerie de liaison série est incohérente, le nombre de variables configurées resterait 6.

Si la description de la variable vecteur de messagerie de liaison série est cohérente, FIPSENSOR sauvegarde cette variable (le nombre de variables configurées passe à 7), et transmet 0x2159 :

7	10	59	0	40	44	50	0	0	14	
59	3	48	5	50	0	0	20	59	1	2
8	50	1	f4	21	59	2	40	7e	5	
0	0	0	30	0	6	70	1	40	0	9
0	0	4	6	10	50	1	f4	90	1	5
6	12	50	1	f4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

La messagerie pour la liaison série est validée, il faut simplement veiller à la présence de l'ID_DAT(0x3000) dans l'arbitre de bus.

3) On va maintenant envoyer la configuration FIP de la variable vecteur de messagerie de configuration opérationnelle, nous allons prendre comme identifieur pour cette variable 0x6000 :

0x2059 (0x60 0x00 0x7 0x70 0x1 0x40 0x0 0x0)

Si la description de la variable vecteur de messagerie de liaison série est incohérente, le nombre de variables configurées resterait 7.

Si la description de la variable vecteur de messagerie de liaison série est cohérente, FIPSENSOR sauvegarde cette variable, et transmet 0x2159 :

8 10 59 0 40 44 50 0 0 14
59 3 48 5 50 0 0 20 59 1 2
8 50 1 f4 21 59 2 40 7e 5
0 0 0 30 0 6 70 1 40 0 0 6
0 0 7 70 1 40 0 0 90 0 4 6
10 50 1 f4 90 1 5 6 12 50
1 f4 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

La messagerie pour la configuration est validée, il faut simplement veiller à la présence de l'ID_DAT(0x6000) dans l'arbitre de bus.

4) On va maintenant envoyer la configuration FIP de la variable réception de la liaison série. Nous allons prendre comme identifieur pour cette variable 0x4000, de type produite, avec une longueur de 126 octets :

0x2059 (0x40 0x00 0x8 0x40 0x7e 0x40 0x0 0x0)

Si la description de la variable vecteur de messagerie de liaison série est incohérente, le nombre de variables configurées resterait 8.

Si la description de la variable vecteur de messagerie de liaison série est cohérente, FIPSENSOR sauvegarde cette variable, et transmet 0x2159 :

9 10 59 0 40 44 50 0 0 14
59 3 48 5 50 0 0 20 59 1 2
8 50 1 f4 21 59 2 40 7e 5
0 0 0 30 0 6 70 1 40 0 0 4
0 0 8 40 7e 40 0 0 60 0 7
70 1 40 0 0 90 0 4 6 10 50
1 f4 90 1 5 6 12 50 1 f4
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0

Pour un "code" 0x8 (voir Tableau 2 : codes associés aux variables), FIPSENSOR assure un transfert aperiodique de cette variable.

5) On va maintenant envoyer la configuration FIP de la variable transmission de la liaison série, nous allons prendre comme identifieur pour cette variable 0x5000, de type consommée, avec une longueur de 126 octets :

0x2059 (0x50 0x00 0x9 0x02 0x7e 0x40 0x1 0xf4)

Si la description de la variable vecteur de messagerie de liaison série est incohérente, le nombre de variables configurées resterait 9.

Si la description de la variable vecteur de messagerie de liaison série est cohérente, FIPSENSOR sauvegarde cette variable, et transmet 0x2159 :

a	10	59	0	40	44	50	0	0	14	
59	3	48	5	50	0	0	20	59	1	2
8	50	1	f4	21	59	2	40	7e	5	
0	0	0	30	0	6	70	1	40	0	4
0	0	8	40	7e	40	0	0	50	0	9
2	7e	40	1	f4	60	0	7	70	1	4
0	0	0	90	0	4	6	10	50	1	f4
90	1	5	6	12	50	1	f4	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

6) Il faut configurer maintenant le reste de l'applicatif : les paramètres de communication de la liaison série, et l'identifiant destinataire du message de réception série.

La configuration sera complète à ce moment.

il suffit de mettre dans un message à destination de 0x6000 le contenu suivant :

message de configuration ("wd" 0x5 0x3 0x19 0xd 0x7e)

le message d'acquittement reçu contient (0x0 0x0 0x0 0x0 0x19 0xd 0x7e)

La liaison série est configurée à 9600 bauds, pas de parité et 1 bit d'arrêt, caractère de fin valide et égal à "cr" (0xd), longueur de bloc 126 octets, mode variable aperiodique.

ensuite on envoie le message de configuration suivant:

message de configuration à destination de 0x6000 ("wf" 0x5 0x59)

le message d'acquittement reçu contient (0x5 0x59)

L'identifiant configuré destinataire du message de réception série est 0x559.

Les paramètres FIP des variables utilisées par FIPSENSOR après cette configuration complète lors d'un mode de communication en variable aperiodique sont les suivants:

<i>identifieurs</i>	<i>longueur</i>	<i>PDU</i>	<i>type</i>	<i>status</i>	<i>transfert message</i>	<i>service</i>	<i>transfert variable</i>
(hexa)	(hexa)	(hexa)					
1459	5	50	produite	pas de raf	sans	periodique	autorise
4000	7e	40	produite	pas de raf	sans	aperiodique	sans
5000	7e	40	consommée	pr(500ms)	sans	aperiodique	sans

Tableau 3 : Configuration de service de l'exemple