



ACTinBOX

CLASSIC-HYBRID

ACTinBOX Classic-Hybrid KNX

ZN1IO-AB46A



Edition 2
Version Classic-Hybrid 1.3

TABLE DES MATIERES

1.	Introduction	4
1.1.	Produit.....	4
1.2.	Objets de communication	4
2.	Sorties.....	6
2.1.	Canaux de volet.....	7
2.1.1.	Type	7
2.1.2.	Times (Temps)	8
2.1.3.	Functions (fonctions).....	9
2.1.3.1.	Status object (Objet d'état) :.....	9
2.1.3.2.	Contrôle précis	10
2.1.3.3.	Scene	10
2.1.3.4.	Block (blocage)	11
2.1.3.5.	Alarm (Alarmes).....	11
2.1.3.6.	reversed Moving (Mouvement inversé).....	12
2.1.3.7.	Direct positionning (positionnement direct).....	13
2.1.3.8.	StartUp (configuration initiale)	13
2.2.	Individual Outputs (Sorties individuelles)	14
2.2.1.	Type	14
2.2.2.	Fonctions	14
2.2.2.1.	Status object (Objet d'état).....	14
2.2.2.2.	Timers (Temporisations)	14
2.2.2.3.	Scènes.....	16
2.2.2.4.	Block (Blocage)	18
2.2.2.5.	Alarm (Alarme)	18
2.2.2.6.	StartUp (configuration initiale)	19
3.	Inputs (Entrées).....	20
3.1.	Push button (Bouton poussoir)	22
3.1.1.	Sending of 0/1 (envoi de 0/1).....	22
3.1.2.	Shutter control (contrôle des volets)	23
3.1.3.	Dimmer control (contrôle de variateur).....	24
3.1.4.	Sendind of a scene (fonction scène)	24
3.1.5.	Configuration supplémentaire	24
3.2.	Switch/sensor (interrupteur / capteur).....	25

3.3.	Sonde de température	26
3.3.1.	Capteur de température	26
3.3.1.1.	Calibrage du capteur de température.....	26
3.3.1.2.	Cycle d'envoi de la température	27
3.3.1.3.	Envoi après un changement de température	27
3.3.1.4.	Protection de température	27
3.3.2.	Capteur de température et thermostat.....	28
3.4.	Détecteur de mouvement :.....	28
4.	Logicals funcions (fonctions logiques).....	28
4.1.	Call (Appel)	29
4.2.	Opérations.....	29
4.2.1.	Description des fonctions – conversion	32
4.3.	Result (Résultat)	34
5.	Objets de communication	36
5.1.	Nomenclature :.....	36

1. INTRODUCTION

1.1.PRODUIT

L'**ACTinBOX CLASSIC-HYBRID** est un actionneur KNX qui combine dans un même dispositif :

- **Sorties 4 x 10A** binaires multifonction (**INDIVIDUELLES ou CANAUX VOLETS ELECTRIQUES**).
- **4 ENTREES** binaires multifonction.
- **1 ENTREE** configurable comme binaire ou analogique pour sonde de température. Permet d'activer un thermostat associé à la sonde de température.
- **1 ENTREE** configurable comme binaire ou analogique/numérique pour le détecteur de mouvement
- **FONCTIONS LOGIQUES** Avancées multi-opération.

Ces 3 blocs (sorties, entrées, et fonctions logiques) travaillent indépendamment ; et peuvent donc tous interagir entre eux comme s'il s'agissait de 3 appareils autonomes connectés au **BUS KNX**.

1.2.OBJETS DE COMMUNICATION

Les Objets de communication de l'**ACTinBOX Classic-Hybrid** sont organisés en quatre groupes principaux :

- Sorties individuelles
- Entrées
- Canaux pour volet
- Fonctions logiques

NOMENCLATURE : Pour faciliter la reconnaissance des objets de communication pendant l'assignation des adresses de groupe, chacun d'eux est nommé en fonction du groupe auquel ils appartiennent, en suivant la règle suivante :

["Groupe auquel il appartient"] "Fonction réalisée"

Les abréviations associées aux différents groupes sont les suivantes :

- Sorties individuelles
 - **[S1] => Sortie 1**
 - **[S2] => Sortie 2**

- **[S3] => Sortie 3**
- **[S4] => Sortie 4**
- Entrées
 - **[E1] => Entrée 1**
 - **...**
 - **[E6] => Entrée 6**
- Canaux de volet
 - **[CA] => Canal A**
 - **[CB] => Canal B**
- Fonctions Logiques

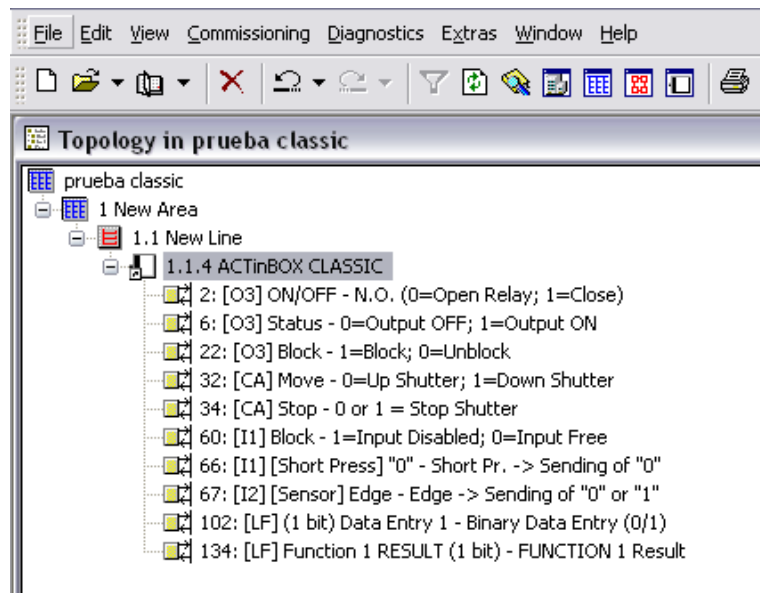


Figure 1

Exemple : voir figure 1.2

- ⇒ [O3] status : *Objet d'état de la sortie 1*
- ⇒ [CA] Stop : *Contrôle d'arrêt du volet électrique du canal B.*
- ⇒ [I1] Block : *Objet bloquant le contrôle sur la sortie 3.*
- ⇒ [LF] Function 1 RESULT (1 bit) : *Objet dans lequel est gardé le résultat (1 bit) de la fonction logique 1.*

2. SORTIES

L'ACTinBOX Classic-Hybrid est doté de **4 sorties binaires de 10A** (voir documentation technique du produit). Ces 4 sorties que l'on nomme sortie 1, sortie 2, sortie 3 et sortie 4, **sont divisées en 2 groupes de 2 sorties chacune**, appelé "Canal":

- **Canal A** : "Sortie 1" et "Sortie 2"
- **Canal B** : "Sortie 3" et "Sortie 4"

Chacun des deux canaux peut être paramétrés indépendamment comme :

- **Sorties individuelles** : chacune des sorties peut être commandée indépendamment. Est utilisé pour le contrôle des charges électriques, des points d'éclairage...
- **Canaux volet électrique** : les deux sorties appartenant, chacune à leur canal, sont contrôlées conjointement. Elles peuvent être utilisées pour le contrôle des volets motorisés (avec ou sans lamelles), auvent...
 - **Dans le cas d'un canal de volet électrique** :
 - ✓ **La sortie 1 (3du canal B)** : est la phase en charge de monter le volet
 - ✓ **La sortie 2 (4 du canal B)** : est la phase en charge de descendre le volet

***Exemple de configuration** : Dans une installation, on veut contrôler un volet électrique et un point d'éclairage.*

Dans ce cas, l'ACTinBOX Classic-Hybrid peut être paramétré comme suit :

Canal A = canal de volet électrique

Canal B = sorties individuelles

Sortie 3 = Normalement fermée

Sortie 4 = Désactivé

2.1.CANAUX DE VOLET

L'ACTinBOX Classic-Hybrid permet d'installer sur ses canaux n'importe quel type de volet ou autres fermetures motorisées. Pour les commandées, sont mis à disposition de l'utilisateur, les objets de contrôle basique "**Move**" et "**stop**", ainsi qu'un ensemble de fonctions additionnelles, chacune avec leurs propres objets de communication.

Le contrôle du volet avec les objets basiques est réalisé comme suit :

- **Monter volet:** Envoi d'un "0" sur l'objet "**move**".
- **Descendre volet:** Envoi d'un "1" sur l'objet "**move**".

***Note I:** Quand l'objet "**move**" reçoit un "0" ou un "1", le volet entre en mouvement, et ne s'arrête qu'en fin de parcours du volet ou lors de la réception d'un nouvel ordre annulant l'antérieur.*

***Note II :** Après la programmation avec l'ETS, l'ACTinBOX Classic-hybrid considère que le volet est complètement monté.*

- **Pour stopper le volet en mouvement :** Envoi d'un "0" ou d'un "1" sur l'objet "**stop**".

2.1.1. TYPE

- **Shutter (no lamella) (volet sans lamelles) :** Il s'agit des volets (ou autres fermetures) d'utilisation habituelles, celles qui ne possèdent qu'un seul mouvement (de monté ou descente).
- **Shutter (with lamella) (volet avec lamelles) :** Ce sont des volets spéciaux possédant des lamelles, dont le mouvement peut être réalisé avec le même moteur. Dans ce cas, l'ACTinBOX Classic-Hybrid se charge du contrôle des mouvements de : **rotation des lamelles** (pour obtenir plus ou moins de luminosité), et **mouvement du volet** (pour monter et descendre le volet). Si ce type de contrôle est sélectionné (volet avec lamelles orientables), l'objet "**stop**" est remplacé par l'objet "**Stop/step**". De telle façon que si le volet se trouve en mouvement lors de la réception d'un "0" ou d'un "1" sur cet objet, le volet s'arrêtera ; alors que si le volet était déjà arrêté lors de la réception d'un "0", les lamelles du volet feront un pas vers le haut (rotation montante des lamelles), ou au contraire, avec un "1", feront un pas vers le bas (rotation descendante des lamelles)

2.1.2. TIMES (TEMPS)

Ils existent deux temps (3 pour le cas des lamelles orientables) à définir pour un fonctionnement correct du Canal

- **Main time (down shutter length) (Temps de parcours du volet) :** c'est le temps que met le volet pour réaliser son parcours complet de montée/descente. Dans le cas où les temps de montée et descente seraient différents, indiquer le temps de descente et sélectionner l'option "**Are total Time is diferente?** (Les temps sont-ils différents?)" et indiquer dans "Total Time Up (Temps de monté)" le temps total de la montée. Il n'est pas nécessaire de graduer ce temps périodiquement, car l'**ACTinBOX Classic-Hybrid** mémorise toujours la position exacte dans laquelle se trouve le volet (même après une coupure de la tension de BUS).

***Note :** Après la programmation via l'ETS, l'**ACTinBOX** considère que le volet est complètement monté.*

- **Secondary time (Lamella Length) (Temps de parcours des lamelles) :** Seulement valide pour les volets ayant des lamelles orientables. Indiquer le temps que tardent les lamelles pour réaliser leur parcours complet.
- **Security time (Temps de sécurité avant le changement de sens) :** C'est le temps de sécurité affecté, par mesure de sécurité, avant le changement de sens du mouvement du volet, pour ne pas surcharger le moteur qui le contrôle. C'est-à-dire, si le volet est en train de monter alors qu'un ordre de descendre lui est envoyé, l'**ACTinBOX Classic-Hybrid** arrêtera le volet un instant (temps défini dans ce champs) avant d'entamer la descente. Il est recommandé un temps d'environ 0.5s (temps défini par défaut).
- **Are total time Up and down diferent ? (les temps de monté et descente sont-ils différents ?) :** Dans le cas de volets très lourd, les temps de montée et descente de ceux-ci peuvent être différents ; dans ce cas, sélectionner cette option et indiquer dans "**total time Up**" (temps total de montée) le temps correspondant à la montée du volet en question.

Exemple de configuration : le volet associé au canal B met 15 secondes pour descendre, et 20 pour monter. Dans ce cas le paramétrage sera celui-ci :

TIMES:	
- Main Time (Down Shutter Length) [x 0.1 sec.]	150
- Secondary Time (Lamella Length) [x 0.1 sec.]	20
- Security Time (Pause to change the movement direction) [x 0.1 sec.]	5
- Are total Time UP and DOWN different?	Yes
Total Time Up [x 0.1s] (Time Down is the param. named above as Main Time)	200

- **Additional time when shutter gets the limit (temps additionnel lorsque le volet arrive en fin de parcours) :** Ce paramètre garantit que le volet atteindra toujours sa fin de parcours. En effet, aux temps de montée et descente définis précédemment, il est possible d'y ajouter un temps supplémentaire de mouvement pendant lequel le moteur continuera sa rotation. Ce temps permet de prévenir les petits désajustements pouvant se produire en fin de parcours du volet.

2.1.3. FUNCTIONS (FONCTIONS)

Chacun des paramètres suivant ajoute des fonctions ou caractéristiques spéciales pour le contrôle de chacun des canaux de volet (A ou B).

2.1.3.1. STATUS OBJECT (OBJET D'ÉTAT) :

Cette fonction met à disposition de l'utilisateur un objet lui indiquant la position du volet à chaque instant.

Il s'agit de l'objet "**Current Position**". C'est un objet de 1 Byte qui indique un pourcentage "%". Ainsi, l'objet vaut 0 (=0%) quand le volet est complètement monté, et vaut 255 (=100%) quand il est complètement descendu. Les autres valeurs indiquent les différentes positions intermédiaires.

Note : L'objet d'état, "**Current position**", a été programmé de telle façon, qu'à chaque fois que le volet entre en mouvement, sa nouvelle position soit envoyée au BUS toutes les secondes, **actualisant ainsi celle-ci en temps réel.**

2.1.3.2. CONTRÔLE PRECIS

Cette fonction permet à l'utilisateur de positionner le volet sur une position choisie, effectuer via un objet de 1 byte "**Positioning**" (Position désirée).

Chaque fois que l'**ACTinBOX Classic-Hybrid** reçoit, sur cet objet, une nouvelle valeur (par exemple 50%), il mettra en mouvement le volet électrique jusqu'à la position correspondante (dans ce cas à la moitié de son parcours)

2.1.3.3. SCENE

Cette fonction permet d'utiliser des scènes (avec un format de 1 Byte) pour contrôler le volet. C'est-à-dire, permet de choisir la position dans laquelle devra se mettre le volet selon la fonction de la scène qui arrive à l'**ACTinBOX Classic-Hybrid** par l'objet "**Scene**".

- **Nombre de scènes :** Ce paramètre permet de déterminer le nombre de scènes utiles, avec un maximum de 5.
- **Scène :** Indique le numéro de la scène qui actionnera le volet.
- **Réaction :** Indique la position sur laquelle se positionnera le volet après réception de l'ordre de la scène indiquée antérieurement.

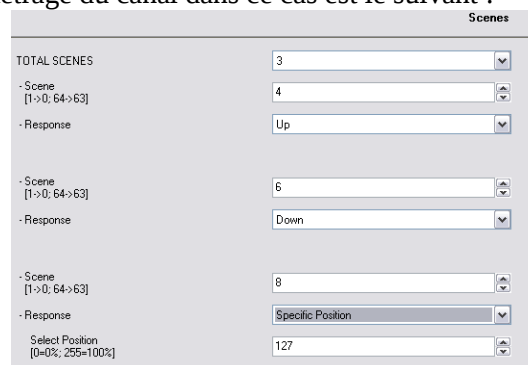
***Exemple :** nous voulons utiliser 4 scènes dans une installation (la 4, la 6, la 8 et la 9). Mais seulement 3 de ces 4 scènes servent pour le contrôle du volet connecté sur l'**ACTinBOX Classic-Hybrid** (scène 4, 6 et 8). Les positions désirées sont les suivantes :*

- Réception scène 4 : monter

- Réception scène 6 : descendre

- Réception scène 8 : moitié du parcours (50%)

Le paramétrage du canal dans ce cas est le suivant :



Scenes	
TOTAL SCENES	3
- Scene [1>0; 64>63]	4
- Response	Up
- Scene [1>0; 64>63]	6
- Response	Down
- Scene [1>0; 64>63]	8
- Response	Specific Position
Select Position [0=0%; 255=100%]	127

Figure 3. Paramétrage des scènes

2.1.3.4. BLOCK (BLOCAGE)

Cette fonction permet de bloquer la sortie. C'est-à-dire désactiver son contrôle (le contrôle ON/OFF et la temporisation). La sortie se bloque lors de l'envoi d'un "1" sur l'objet "**Block**"; et se débloque avec l'envoi d'un "0".

***Note :** La fonction alarme est la seule à avoir une priorité plus grande que la fonction blocage. C'est-à-dire, que même en étant en position blocage, si une alarme survient, la sortie se positionnera en position indiquée par la Fonction Alarme. A la fin de l'alarme, la sortie revient en position blocage.*

2.1.3.5. ALARM (ALARME)

Cette fonction a été créée pour le cas où les **ACTinBOX Classic-Hybrid** devraient réagir face à des situations d'alarme.

Devant une alarme, cette fonction oblige la sortie à se mettre dans la position (ON, OFF ou intermittence) choisie par paramètre ; et après l'alarme, la bloque, empêchant ainsi son contrôle, jusqu'à la fin de l'alarme.

- **Nombre d'alarmes :** Ce paramètre détermine l'**activation de 1 ou 2 alarme(s)**. Dans le cas de choisir 2 alarmes, chacune d'elles se contrôle indépendamment avec les objets de communication ("**alarm 1**" pour la 1^{er} et "**alarm 2**" pour la 2nd)

***Note :** L'alarme 1 a la priorité sur l'alarme 2. C'est-à-dire, si le Canal se trouve dans l'état d'alarme 2, et se produit l'alarme n°1, le volet se mettra en position d'alarme 1, et ne reviendra en position alarme 2 seulement lorsque l'alarme 1 cessera. Alors que si le volet se trouve déjà en alarme 1 et survient une alarme 2, le système continuera dans la position de l'alarme 1.*

- **Trigger value (valeur déclenchant l'alarme) :** Ce paramètre détermine la valeur à appliquer pour démarrer l'alarme. C'est-à-dire, il se produira une alarme quand sur l'objet "alarm" sera détecté la valeur défini dans ce paramètre (celle qui n'active pas la fonction s'appelle **Valeur Passive**).
- **Cyclical monitoring (Contrôle cyclique) :** il est parfois nécessaire de savoir, à chaque instant, qu'aucune alarme ne se soit produite. Pour cela, capteur qui envoi la valeur de déclenchement devra envoyer continuellement la **Valeur Passive** tant que le système ne se trouvera pas en état d'alarme. Dans ce cas particulier l'option "cyclical Monitoring" doit être activée. De cette façon lorsque l'**ACTinBOX Classic-Hybrid** ne recevra pas la Valeur Passive pendant un temps défini dans "**cycle time**" (temps de cycle), l'alarme s'activera automatiquement.

- **Cycle time (temps de cycle) :** Temps nécessaire entre la réception de deux Valeur Passive avant le déclenchement de l'alarme.

***Note :** il est recommandé de choisir un temps supérieur au double du cycle d'envoi du capteur, au cas où certains messages se perdraient.*

- **Reponse (Réaction) :** Ce paramètre détermine quelle sera la réaction de la sortie (indique la position sur lequel se positionnera le volet) au déclenchement de l'alarme. (Dans le cas de choisir "**specific position**" (position spécifique), il faut se rappeler que : 0=0% et 255=100%).
- **Désactivation :** il existe 2 méthodes pour désactiver l'alarme :
 - ✓ **Normal :** Consiste à envoyer sur l'objet "**Alarm**" la Valeur Passive (valeur opposée à celle de déclenchement)
 - ✓ **Frozen (blocage) :** Cette fonction consiste à appliquer la méthode normale mais ensuite envoyer un "1" sur l'objet "**Unfreeze alarm**". Cette seconde méthode permet, même si la situation d'alarme s'est terminée, de conserver le blocage de la sortie jusqu'à l'envoi d'un ordre sur son objet de communication.
- **Etat final :** Ce paramètre détermine ce que fera la sortie à la fin de l'alarme (current position (rester dans sa position), Up (monter), down (descendre), last position (Dernière position avant le déclenchement de l'alarme)).

2.1.3.6. REVERSED MOVING (MOUVEMENT INVERSÉ)

Cette fonction permet d'inverser le contrôle des volets. C'est-à-dire, monter les volets avec un "1", et les descendre avec un "0".

Cette fonction utilise l'objet de communication "**Reversed moving**" (Mouvement inversé). Ce type de contrôle est compatible avec le contrôle normal. Ainsi, avec l'objet de "**Move**" est réalisé le contrôle normal et avec l'objet "**Reversed moving**" le contrôle inversé.

Ceci est utile, par exemple, s'il est souhaité effectuer avec un seul et même contrôle, une extinction générale des éclairages et la fermeture des volets avec la valeur "0". Dans ce cas, sera envoyé un "0" sur les objets "**ON/OFF**" du contrôle de l'éclairage et les objets "**mouvement inversé**" du contrôle des volets.

2.1.3.7. DIRECT POSITIONNING (POSITIONNEMENT DIRECT)

Cette fonction permet de positionner les volets sur une position fixe choisie, utilisant des objets de 1 bit.

A la réception d'un "1" sur un de ces objets ("**direct positioning**" ou "**direct positioning 2**"), le volet se positionnera sur la position prédéterminée par paramètre.

- **Total direct positionings (nombre de position)** : Ce paramètre permet de choisir s'il sera utilisé 1 ou 2 position(s).
- **Select position** : Ce paramètre permet de d'indiquer la position choisie. (Rappel 0 = 0% et 255 = 100%).
- **New positions saving (nouvelle sauvegarde de position)** : Ce paramètre détermine s'il sera autorisé la sauvegarde de nouvelles positions du volet. Lors de l'activation de cette option apparaissent deux nouveaux objets :
 - ✓ **Save position**
 - ✓ **Save position 2**

Pour enregistrer de nouvelles positions, il faut envoyer un "1" sur ces objets.

2.1.3.8. STARTUP (CONFIGURATION INITIALE)

Cette fonction permet de choisir ce que devra faire le volet à son initialisation (au retour de la tension de BUS sur l'**ACTinBOX Classic-Hybrid** ou après le téléchargement d'un programme via l'ETS).

- **Initial position (Position initiale)** : Ce paramètre définit la position du volet lors de son actualisation à choisir entre :
 - ✓ **Current position (position actuelle)** : Reste dans la position dans laquelle il se trouve.
 - ✓ **Up** : Monter le volet
 - ✓ **Down** : Descendre le volet
 - ✓ **Specific position** : Le mettre dans la position spécifiée dans "select position".
- **Update (mise à jour)** : Le paramètre Update permet d'envoyer au reste des dispositifs la position dans lequel se trouve le volet (après une coupure électrique ou chargement programme). Comme certains appareils mettent plus de temps que d'autres à s'initialiser, il est possible de **retarder** l'envoi du signal, afin de s'assurer que tous les appareils soient prêts pour recevoir le message, avec le paramètre "Start-Up sending delay".

Note : l'envoi de l'état initial est toujours réalisé à travers l'objet "current position"

2.2.INDIVIDUAL OUTPUTS (SORTIES INDIVIDUELLES)

Dans ce cas, chacune des deux sorties du canal contrôlera des dispositifs différents et indépendants. Si vous ne voulez pas utiliser la sortie, il est possible de la désactiver.

2.2.1. TYPE

Il faut indiquer si la sortie est du type "**Normally open**" (normalement ouvert) ou "**normally closed**" (normalement fermé). Le comportement du ON et OFF sera le suivant selon ce paramètre :

Normalement ouvert : ON => Fermer relais ; OFF => Ouvrir Relais

Normalement fermer : ON => Ouvrir relais ; OFF => Fermer Relais

Note : Lorsque l'on parle des sorties individuelles **ON = 1** et **OFF = 0**. Cela signifie que chaque fois qu'il sera fait référence qu'un objet envoie ou reçoit un "**ON**", signifie qu'il envoie ou reçoit la valeur "1" (et vice versa). Dans certaine occasion le ON est appelé Allumage, et le OFF extinction. Le contrôle normal de la sortie est effectué sur l'Objet de Communication "**ON/OFF**".

2.2.2. FONCTIONS

Chacun des paramètres suivants ajoute des fonctionnalités ou caractéristiques spécifiques au contrôle de chacune des Sorties Individuelles.

2.2.2.1. STATUS OBJECT (OBJET D'ÉTAT)

L'objet d'état de communication "**status**" renvoie l'état dans lequel se trouve la sortie à chaque changement d'état.

⇒ Quand la sortie est à ON, cet objet vaut "**1**".

⇒ Quand la sortie est à OFF, cet objet vaut "**0**".

Chaque fois que l'état de la sortie change, la nouvelle valeur de cet objet est envoyée au BUS.

2.2.2.2. TIMERS (TEMPORISATIONS)

Il est possible temporiser le contrôle de la sortie.

Il existe 2 types de contrôle temporisés :

- **Simple timer (Temporisation simple)** : Fonction réalisée via l'objet "timer"
- **Flashing (intermittence)** : Fonction réalisée via l'objet "flashing"

Ces contrôles temporisés sont indépendants les uns des autres, et indépendant du contrôle normal "ON/OFF", grâce à la présence de trois objets de communication différents.

Par exemple, si un **ON** est envoyé sur l'objet "timer", commence un ON temporisé sur la sortie. Si avant de terminer le ON Temporisé est envoyé un OFF sur l'objet "ON/OFF", la sortie s'éteindra et la temporisation annulée.

Les paramètres que l'on trouve sont :

- **Simple timer (temporisation simple)** : Il s'agit d'une temporisation normale que l'ACTinBOX Classic-Hybrid applique sur la sortie à la réception d'un ON ou d'un OFF par l'objet "Timer".
 - ✓ **ON delay (Retard à l'allumage)** : Ce paramètre permet de déterminer la temporisation à respecter entre l'ordre de ON (sur l'objet de communication "timer") et l'allumage effectif (ON) de la sortie. Ce temps se mesure en dixième de seconde (par exemple, si un retard de 2.5 secondes est souhaité, il faudra choisir la valeur 25 dans le paramètre). Un "0" indique qu'aucun retard ne sera appliqué.
 - ✓ **OFF delay (retard à l'extinction)** : Ce paramètre permet de déterminer la temporisation à respecter entre l'ordre de OFF (sur l'objet "timer") et l'extinction effective (OFF) de la sortie. (Son fonctionnement est le même que le retard à l'allumage).
 - ✓ **ON duration (Temps allumé)** : Cette fonction permet de déterminer le temps pendant lequel la sortie restera à ON avant de revenir à l'état OFF. Un "0" indique que la durée est illimitée (qu'aucune temporisation ne sera appliquée sur le paramètre de "durée de l'allumage").

Note: Les fonctions retard et durée sont détaillées à continuation :

- Quand est envoyé un "1" sur l'objet "Timer", l'ordre d'allumage est envoyé sur la sortie en lui appliquant le retard à l'allumage et la durée de l'allumage.

- Quand est envoyé un "0" sur l'objet "Timer", l'ordre d'extinction est envoyé sur la sortie en lui appliquant le retard à l'extinction.

- ✓ **Multiply** : Cette fonction consiste en l'accumulation d'une temporisation autant de fois que sera reçu un "1" ou un "0" sur l'objet de communication "Timer".

Note : Le fonctionnement de la multiplication est détaillé à continuation :

- Sans multiplication : Si pendant un ON temporisé est reçu un "1" sur l'objet de communication "**timer**", l'**ACTinBOX Classic-Hybrid** recommencera à compter le nouveau ON temporisé.

- Avec multiplication : Si pendant un ON Temporisé est reçu un "1" sur l'objet de communication "**timer**", l'**ACTinBOX Classic-Hybrid** additionnera le temps précédent avec le nouveau temps de timer, soit un timer deux fois plus long que le premier, si un autre "1" est envoyé, le temps sera trois fois plus long etc...

Même fonctionnement pour le OFF temporisé.

- **Flashing (intermittence):** La fonction intermittence permet de réaliser une séquence sur la sortie, du type ON-OFF-ON-OFF...
 - ✓ **On duration (temps allumé):** Ce paramètre détermine le temps pendant lequel la sortie restera en position ON à chaque répétition.
 - ✓ **Off duration (temps éteint):** Ce paramètre détermine le temps pendant lequel la sortie restera en position OFF à chaque répétition.
 - ✓ **Repetitions:** Ce paramètre définit le nombre de répétition à exécuter. Pour un nombre de répétitions illimitées, choisir la valeur "0".
 - ✓ **Status after last repetition (Etat après la dernière répétition) :** Ce paramètre permet de définir l'état dans lequel se positionna la sortie après la dernière répétition.

Note : L'intermittence est activée lors de l'envoi d'un "1" sur l'objet de communication "**flashing**", et s'arrête en lui envoyant un "0".

2.2.2.3. SCÈNES

Cette fonction permet de contrôler la sortie via des objets de communication "**scene**" utilisant pour cela des scènes BUS (type 1 byte).

- **Scene number (numéro de la scène) :** Ce paramètre permet de sélectionner jusqu'à un maximum de 5 scènes.
 - ✓ **Scène:** Indique le numéro de la scène devant laquelle réagira la sortie.
 - ✓ **Reponse (réaction):** Indique la position dans laquelle se mettra la sortie (ON ou OFF) quand arrivera ce numéro de scène via l'objet de communication "**scene**".

Exemple : il est souhaité utiliser 4 scènes dans une installation (la 4, la 6, la 8 et la 9).

Mais il est seulement nécessaire que la sortie de l'**ACTinBOX** se mette dans une position concrète (ON ou OFF) pour les trois premières (la 4, la 6 et la 8).

Les positions souhaitées sont :

- Pour la scène 4 : ON

- Pour la scène 6 : ON

- Pour la scène 8 : OFF

Pour cela, il faut paramétrer cette sortie de l'**ACTinBOX** de la façon suivante :

TOTAL SCENES	3
- Scene [1->0; 64->63]	4
- Response	ON
- Scene [1->0; 64->63]	6
- Response	ON
- Scene [1->0; 64->63]	8
- Response	OFF

Figure 4 : Paramétrage des scènes

2.2.2.4. BLOCK (BLOCAGE)

Grâce à cette fonction, il est possible de bloquer la sortie. C'est-à-dire, désactiver son contrôle (le contrôle ON/OFF et le timer). La sortie se bloque en envoyant un "1" sur l'objet "block" : et se débloque en envoyant un "0".

***Note :** La seule fonction prioritaire sur la fonction bloque est la fonction alarme. C'est-à-dire, si étant en position de blocage survient une alarme, la sortie se positionnera dans la position déterminée par la fonction alarme. En fin d'alarme, la sortie revient en position bloquée.*

2.2.2.5. ALARM (ALARME)

Cette fonction est faite pour les cas où l'**ACTinBOX Classic-Hybrid** devrait réagir devant des situations d'alarme.

Lors d'une alarme, cette fonction oblige la sortie à se positionner dans la position (ON, OFF ou intermittence) choisie par paramètre ; L'alarme bloque la sortie, empêchant son contrôle, jusqu'à la fin de l'alarme.

- **Trigger value (valeur de déclenchement) :** Ce paramètre détermine la valeur qui activera l'alarme. C'est-à-dire, se produira une alarme quand l'objet "**alarm**" rencontrera la valeur de déclenchement définie pas paramètre. La valeur opposée (celle qui n'enclenche pas l'alarme) s'appelle **ValeurPassive**.
- **Cyclical monitoring (Contrôle cyclique) :** Il est parfois nécessaire de connaître, à chaque instant, qu'aucune alarme ne se soit déclenchée. Pour cela, le capteur qui envoie la valeur de déclenchement à l'**ACTinBOX Classic-Hybrid**, doit envoyer en continue la **ValeurPassive** quand il n'est pas en position d'alarme. C'est dans ce type de cas que doit être activé l'option "Contrôle cyclique". De cette manière, si l'**ACTinBOX Classic-Hybrid** ne reçoit pas la **ValeurPassive** pendant le temps choisi dans "**Cycle Time**", l'alarme s'activera automatiquement.

***Note :** Il est recommandé de choisir un temps supérieur au double du cycle d'envoi du capteur, au cas où un message se perdrait.*

- **Response (Réaction) :** Ce paramètre détermine quelle sera la réaction de la sortie (ON, OFF ou intermittence) au déclenchement de l'alarme. La réaction de type Intermittence génère, lorsque l'alarme est déclenchée, une séquence ON-OFF-ON-OFF..., dont la durée de chaque état est à déterminer dans les paramètres suivants :
 - ✓ **ON Duration (durée du ON) :** Détermine la durée du ON à chaque répétition.

- ✓ **OFF Duration (durée du OFF):** Détermine la durée du OFF à chaque répétition.
- ✓ **Repetitions:** Nombre de répétition de la fonction à effectuer. Pour un nombre de répétitions infinies, choisir la valeur "0".
- ✓ **Status after last repetition (Etat après la dernière répétition):** Permet de déterminer l'état dans lequel se positionnera la sortie après la dernière répétition.
- **Désactivation :** Il existe 2 méthodes pour désactiver l'alarme :
 - ✓ **Normale :** Consiste à envoyer sur l'objet "**alarm**" la **ValeurPassive** (valeur opposée à celle qui déclenche l'alarme)
 - ✓ **Frozen (maintient alarme):** Cette fonction utilise la même méthode que l'antérieure (Méthode Normale), avec en plus, l'obligation d'envoyer un "1" sur l'objet "**Unfreeze alarm**". Cette seconde méthode permet de, même si la situation d'alarme a conclu, conserver en état bloqué la sortie, et ce, jusqu'à l'envoi d'un contre ordre sur l'objet "unfreeze alarm" avec un "1".
- **Etat final :** Ce paramètre permet de choisir l'état dans lequel se positionnera la sortie à la fin de l'alarme (No change (rester dans son état actuel), ON, OFF (se mettre en position ON / OFF), last (position qu'il avait avant le déclenchement de l'alarme)).

2.2.2.6. STARTUP (CONFIGURATION INITIALE)

Cette fonction permet de choisir la position dans laquelle se mettra la sortie (ON ou OFF) à l'initialisation, c'est-à-dire, au retour de la tension du BUS sur l'**ACTinBOX Classic-Hybrid**, ou après le téléchargement d'une programmation depuis l'ETS.

- **Last saved position (dernière position enregistrée):** L'option "**dernière position enregistrée**" signifie qu'après une coupure de tension sur le BUS, la sortie reviendra à la position d'avant la coupure. Par contre, **après une programmation** via l'ETS, la sortie se mettra en position OFF, c'est-à-dire avec le relais ouvert si la sortie est normalement ouverte (en vice-versa)
- **ON :** la sortie se mettra en position ON au retour de la tension de BUS et après la programmation via l'ETS.
- **OFF :** la sortie se mettra en position OFF au retour de la tension de BUS et après la programmation via l'ETS.
- **Update (mise à jour):** Ce paramètre permet d'envoyer sur le BUS l'état initial de la sortie, pour actualiser les appareils de l'installation.

- ✓ **Start-up sending delay (delai d'envoi de l'état) :** Comme certains appareils de l'installation peuvent mettre un certain temps pour s'initialiser, il est possible de retarder l'envoi de l'état initial sur le BUS afin de s'assurer que tous les appareils soient prêts à recevoir le message. L'envoi de l'état initial s'effectue toujours via l'objet de communication "**Status**".

3. INPUTS (ENTRÉES)

L'ACTinBOX Classic-Hybrid dispose de 6 Entrées :

- **Entrées binaires pour contacts libres de potentiel :** Entrées 1, 2, 3, 4, 5 et 6. Chacune d'elle peut être connectée à un bouton poussoir ou un interrupteur/capteur.
- **Entrées analogique pour sonde de température :** l'entrée 5 peut être connectée à une sonde de température et est, en plus, dotée d'un thermostat.
- **Entrée analogique/Numérique pour détecteur de mouvement :** l'entrée 6 peut être connectée à un détecteur de mouvement.

Le fait d'unir physiquement une entrée avec le "**commun**" implique un Contact fermé sur cette entrée. Et les séparer implique le contraire (contact ouvert).

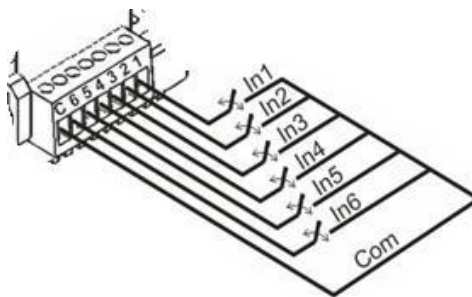


Figure 5 : Connecteur avec entrées binaires

Sur une entrée de type Bouton Poussoir, il est possible de réaliser les fonctions suivantes :

- **Sending of 0/1 (Envoi de 0/1):** Envoi sur le BUS de la valeur (1bit) "0" ou "1".
- **Shutter control (contrôle volet):** Envoi sur le BUS de l'objet correspondant pour mettre en mouvement / arrêter un volet (ou plusieurs).
- **Dimmer control (contrôle variateur):** Envoi sur le BUS de l'objet correspondant pour un variateur (ou plusieurs).
- **Sending of scene (envoi d'une scène):** Envoi sur le BUS d'une scène au format 1byte (exécution et sauvegarde).

Il est possible de choisir que l'**ACTinBOX Classic-Hybrid** réalise une de ces fonctions devant une pression courte et, indépendamment, qu'il en réalise une autre devant une pression longue.

***Exemple :** L'entrée 3 peut contrôler un volet avec une pression courte, et exécuter une scène devant une pression longue.*

Ceci permet à l'**ACTinBOX Classic-Hybrid** de contrôler jusqu'à 12 fonctions indépendantes sur ses entrées.

Sur une entrée de type **CAPTEUR**, est réalisée la fonction suivante :

- **Envoi de 0/1 :** pour chaque front, montant ou descendant, choisir s'il sera envoyé sur le BUS un "0", un "1", ou une commutation de "0" et de "1".

Sur l'entrée type **SONDE DE TEMPERATURE** est réalisée la fonction suivante :

- **Envoi de température**
- **Envoi de température et contrôle thermostatique**

Sur l'entrée type **DETECTEUR DE MOUVEMENT** est réalisée la fonction suivante :

- **Envoi de 0/1 :** Pour chaque état de détection ou non détection, il est possible de choisir si sera envoyé un "1" ou un "0" au BUS.
- **Envoi de scène :** Pour chaque état de détection ou non détection, il est possible de choisir l'envoi d'un numéro de scène au BUS.
- **Réception scène :** Il est possible de recevoir une scène via le BUS afin d'activer/désactiver la détection dans un canal.
- **Envoi du niveau de luminosité :** Permet d'envoyer périodiquement le niveau de luminosité détecté, qui va de 0 à 100.
- **Court-circuit :** Indique s'il s'est produit un court-circuit entre l'entrée 6 et le commun, empêchant ainsi son bon fonctionnement.
- **Circuit-ouvert :** Indique si l'entrée 6 et le commun sont en circuit ouvert, dans ce cas le capteur ne fonctionnera pas correctement.

3.1.PUSH BUTTON (BOUTON POUSSOIR)

Un bouton poussoir connecté sur une entrée consiste en un mécanisme, qui, en conditions normales, maintient ses contacts ouverts. Les contacts sont fermés pendant l'actionnement du bouton poussoir et reviennent à leur état de repos après. Cette action s'appelle "**pression**", et le temps que dure celle-ci s'appelle "**temps de pression**".

- Short press (**pression courte**)
- Long press (**pression longue**)

Sont en générale utilisés comme boutons poussoir les mécanismes de pression conventionnels comme les touches ou les claviers.

***Note :** L'ACTinBOX n'accepte pas les boutons poussoir Normalement Fermés, c'est-à-dire, connectés de telles manières qu'en conditions normales (de repos), ils aient leurs contacts fermés.*

3.1.1. SENDING OF 0/1 (ENVOI DE 0/1)

Cette fonction génère l'envoi sur le **BUS** d'un objet de 1 bit.

- **Response (réaction) :** Cette fonction détermine la valeur qui sera envoyée "0", "1" ou une commutation alternative "0" et "1".
- **Cycle response sending (envoi cyclique de la valeur) :** Ce paramètre détermine s'il sera envoyé périodiquement la valeur "0", "1" ou les deux (si "Always" (**Toujours**) est sélectionné).
- ✓ **Cycle time (cycle d'envoi) :** Ce paramètre définit l'intervalle de temps entre l'envoi de deux messages.

***Note:** les liens internes qui existaient dans l'ACTinBOX Classic ont été éliminés car le nouveau firmware 5.0 supporte le Full Duplex.*

3.1.2. SHUTTER CONTROL (CONTRÔLE DES VOLETS)

Cette fonction a comme résultat l'envoi sur le BUS d'un objet de 1 bit pour le contrôle des volets.

- **Response :** Ce paramètre permet de déterminer si l'objet sera utilisé pour le contrôle de :
 - ✓ **Up (Monter)**
 - ✓ **Down (descendre)**
 - ✓ **Up/down switched (monter/descendre commuté) :**
permet le contrôle du volet avec une seule entrée.
 - ✓ **Stop shutter (Arrêter volet)**

***Note :** Ce dernier mode "Arrêter volet" inclut les 3 options de "step Up", "step down", "step switched" (un pas vers le haut, vers le bas et commuté) pour le contrôle des lamelles ; mais s'il n'y a pas de lamelles orientables, ces trois contrôles servent uniquement pour arrêter le volet.*

***Note:** les liens internes qui existaient dans l'ACTinBOX Classic ont été éliminés car le nouveau firmware 5.0 supporte le Full Duplex*

3.1.3. DIMMER CONTROL (CONTRÔLE DE VARIATEUR)

Cette fonction a comme résultat l'envoi sur le **BUS** d'un objet de contrôle de variateurs.

- **Response (réaction)** : En fonction de l'option choisie, cet objet peut être :
 - ✓ **Contrôle ON/OFF**:
 - ⇒ **Light ON (allumer)** : Envoi un "1"
 - ⇒ **Light OFF (éteindre)** : Envoi un "0"
 - ⇒ **Light ON/OFF switched** : Envoi "0" et "1" alternativement.
 - ✓ **Contrôle de régulation** : Selon l'option choisie, il envoie un objet de 4 bits correspondant.
- **Dimming step (pas de régulation)** : Si une des options de contrôle de régulation est choisie, il est nécessaire de définir ce paramètre qui correspond à l'augmentation ou diminution de la luminosité qu'effectuera le **VARIATEUR** à chaque pas.

3.1.4. SENDIND OF A SCENE (FONCTION SCÈNE)

Cette fonction s'utilise pour contrôler une scène du **BUS** avec l'entrée.

- **Reponse (réaction)** : Ce paramètre permet de déterminer si l'action à réaliser est une exécution ou enregistrement d'une scène (ordre à tous les appareils qu'ils exécutent la scène émise).
- **Scène** : Détermine le Numéro de la scène à exécuter/enregistrer.

***Note:** les liens internes qui existaient dans l'ACTinBOX Classic ont été éliminés car le nouveau firmware 5.0 supporte le Full Duplex*

3.1.5. CONFIGURATION SUPPLÉMENTAIRE

Cette fonction permet de paramétrer les différentes pressions :

- **Threshold time (temps de pression)** : Etabli un laps de temps qui permet de distinguer la "pression courte" de la "pression longue".
- **Reponse delay (retard)** : Il est possible d'ajouter une temporisation avant l'envoi de l'objet associé à l'action effectuée depuis l'entrée (pression courte ou longue). Une valeur "0" indique que l'envoi sera immédiat (sans temporisation).

Exemple : *Devant une pression courte, l'ACTinBOX Classic-Hybrid attendra le temps défini dans " Reponse delay (short press) " avant d'envoyer l'objet correspondant sur le BUS.*

- **Block (blocage) :** Ce paramètre active l'objet "**block**". Cet objet s'utilise pour désactiver l'entrée.
 - ✓ A la réception d'un "1" sur cet objet, l'ACTinBOX Classic-Hybrid bloque tous les ordres venant de l'entrée.
 - ✓ A la réception d'un "0" sur cet objet, l'entrée se débloque, mais tous les ordres envoyés depuis l'entrée, pendant son état de blocage, seront perdus.

3.2.SWITCH/SENSOR (INTERRUPTEUR / CAPTEUR)

Un interrupteur/capteur connecté sur une entrée consiste en un mécanisme dont son état normal (de repos) peut être en contact ouvert ou fermé, et ne revenant pas à son état initial automatiquement.

Le passage d'un état à l'autre s'appelle "**edge**" ("**front**"):

- **Falling edge (front descendant) :** Passage d'un contact fermé à un contact ouvert.
- **Rising edge (front montant) :** Passage d'un contact ouvert à un contact fermé.

Sont d'habitude utilisés comme interrupteur /capteur les contacts de sortie libre de potentiel des capteurs conventionnels et des relais, en général.

Le fait de sélectionner une entrée comme "**interrupteur/Capteur**", un objet de 1 bit "**[sensor] edge**" apparaît et sera envoyé sur le **BUS** à chaque fois qu'un front montant et/ou descendant sera détecté sur l'entrée, selon les paramètres définis.

- **Rising edge :** Ce paramètre détermine la valeur qui sera envoyée devant un front montant sur l'entrée.
- **Falling edge :** Ce paramètre détermine la valeur qui sera envoyée devant un front descendant sur l'entrée.
- **Sending of "0" delay (retard du "0") :** Ce paramètre détermine le temps que devra attendre l'ACTinBOX Classic-Hybrid avant d'envoyer la valeur "0" sur l'objet de communication "**[sensor] edge**"
- **Sending of "1" delay (retard du "1") :** Ce paramètre détermine le temps que devra attendre l'ACTinBOX Classic-Hybrid avant d'envoyer la valeur "1" sur l'objet de communication "**[sensor] edge**"
- **Periodical sending of "0" (envoi cyclique du "0") :** Ce paramètre détermine le cycle d'envoi de la valeur "0". C'est-à-dire, quand l'objet "**[sensor] edge**" vaudra "0", il enverra cette valeur sur le **BUS** périodiquement. Si l'envoi cyclique n'est pas nécessaire, introduire la valeur "0".

- **Periodical sending of "1" (envoi cyclique du "1") :** Ce paramètre détermine le cycle d'envoi de la valeur "1". C'est-à-dire, quand l'objet "[sensor] edge" vaudra "1", il enverra cette valeur sur le **BUS** périodiquement. Si l'envoi cyclique n'est pas nécessaire, introduire la valeur "0".
- **Block :** Ce paramètre active l'objet "block". Cet objet sert à désactiver l'entrée :
 - A la réception d'un "1" sur cet objet, l'**ACTinBOX Classic-Hybrid** ne répondra plus aux différents fronts qui pourraient surgir sur l'entrée.
 - A la réception d'un "0" sur cet objet, l'entrée de l'**ACTinBOX Classic-Hybrid** redevient sensible aux différents fronts qui pourraient surgir sur l'entrée (tous les ordres envoyés pendant son état de blocage sont perdus).

***Note:** les liens internes qui existaient dans l'**ACTinBOX Classic-Hybrid** ont été éliminés car le nouveau firmware 5.0 supporte le Full Duplex*

3.3.SONDE DE TEMPÉRATURE

Une **sonde de température** permet de mesurer la température réelle de l'endroit désiré. Cette mesure peut être envoyée périodiquement via un objet de communication.

En plus de la sonde de température, le programme d'application de l'**ACTinBOX Classic-Hybrid** permet d'activer un **thermostat**.

Les options pouvant être choisies pour cette entrée sont :

- Capteur de température
- Capteur de température et thermostat

3.3.1. CAPTEUR DE TEMPÉRATURE

La sélection de cette option permet de paramétrer le capteur de température.

3.3.1.1. CALIBRAGE DU CAPTEUR DE TEMPÉRATURE

Permet de calibrer le capteur de température, en prenant comme référence la température mesurée par celle-ci, et y ajoutant ou soustrayant un paramètre introduit dans l'ETS.

Ceci signifie, par exemple que, si l'utilisateur dispose d'une sonde de température très précise, il lui sera possible de prendre comme référence cette température mesurée.

Mesure indiquée par la sonde de température	x °C
---	------

Mesure indiquée par le thermomètre de précision	y °C
Différence positive entre les mesures	$x - y = z$ °C
Paramètre positif de calibration	z °C
Différence négative entre les mesures	$x - y = -z$ °C
Paramètre positif de calibration	- z °C

3.3.1.2. CYCLE D'ENVOI DE LA TEMPÉRATURE

Permet de définir un cycle d'envoi de la valeur de la température, enregistrée par le capteur de l'**ACTinBOX Classic-Hybrid**, sur le BUS via l'objet de communication de **Température Actuelle**.

3.3.1.3. ENVOI APRÈS UN CHANGEMENT DE TEMPÉRATURE

Permet d'envoyer la valeur de la température, mesurée par le capteur de l'**ACTinBOX Classic-Hybrid**, sur le BUS, via l'objet de communication de Température réelle, quand se produit une variation de température supérieure à celle paramétrée.

3.3.1.4. PROTECTION DE TEMPÉRATURE

Il est possible de choisir une **Protection de Température** pour la surchauffe, congélation, ou les deux.

S'il est choisi la **protection de surchauffe**, un objet de communication de **1 bit** apparait, par défaut il prend la valeur 0. Quand la température mesurée par la sonde de température dépasse celle paramétrée dans l'option **Température de Surchauffe**, un 1 est envoyé sur l'objet de communication associé.

S'il est choisi la **protection de Congélation**, un objet de communication de **1 bit** apparait, par défaut il prend la valeur 0. Quand la température mesurée par la sonde de température passe sous celle paramétrée dans l'option **Température de Congélation**, un 1 est envoyé sur l'objet de communication associé.

3.3.2. CAPTEUR DE TEMPÉRATURE ET THERMOSTAT

Lorsque cette option est sélectionnée, il est possible de paramétrer le capteur de température et le thermostat Zennio.

Les options pour le capteur de température sont les mêmes que celles indiquées dans le chapitre précédent.

Les possibilités du thermostat Zennio, ainsi que son paramétrage sont disponibles dans le document **Climat 1. Thermostat Zennio**.de notre page internet.

3.4.DÉTECTEUR DE MOUVEMENT :

Toute l'information sur le paramétrage de l'entrée 6 comme détecteur de mouvement se trouve dans le manuel de celui-ci téléchargeable depuis notre page internet.

4. LOGICALS FUNCIONS (FONCTIONS LOGIQUES)

Cette section de l'**ACTinBOX Classix-Hybrid** permet de réaliser des opérations en logique binaire avec des données venant du BUS, et renvoyer le résultat via d'autres objets de communication, spécialement activés pour cet effet dans l'actionneur. Les données utilisées par les fonctions logiques peuvent provenir de 2 endroits différents :

- Du **BUS**, via les **objets de communication** spécialement créés pour ces fonctions.
- De **variables internes** dans lesquelles sont gardés les résultats partiels des opérations intermédiaires.

Il faut configuré les paramètres correspondants dans la section **Fonctions Logiques** :

- **Fonctions logiques utilisées** : Il est possible d'utiliser jusqu'à 5 fonctions logiques différentes, indépendantes les unes des autres. Pour les utiliser il est nécessaire de les activer.
 - Fonction 1....5
- **Total data entry objects (nombre total d'objet entrant)** : Définir le nombre d'objets de données d'entrée de chaque type, qui seront utilisés entre toutes les fonctions logiques.
 - **1bit (16 objets disponibles)** : Ici il faut définir le nombre d'objet de 1 bit qui seront utilisés entre toutes les opérations des fonctions logiques.
 - **1 byte (8 objets disponibles)** : Ici il faut définir le nombre d'objet de 1 byte qui seront utilisés entre toutes les opérations des fonctions logiques.

- **2 bytes (8 objets disponibles) :** Ici il faut définir le nombre d'objet de 2 bytes qui seront utilisés entre toutes les opérations des fonctions logiques.

***Note I :** Est également mit à disposition de l'intégrateur :*

- 16 variables de 1 bit

- 8 variables de 1 byte

- 8 variables de 2 bytes

Variables internes utilisées pour stocker les résultats partiels des opérations

***Notes II :** A partir du moment que le nombre des objets des données d'entrée ne sera pas définis par paramètre, les objets de communications associés à ceux-ci n'apparaîtront pas dans l'ETS.*

***Note III :** il est fortement recommandé d'activer quelques objets supplémentaires, même s'ils ne sont pas utilisés dans un premier temps, car, par la suite, le fait d'activer d'autre objet supprime toutes les associations d'adresse de groupe qu'il peut y avoir sur tous les objets de données d'entrée. Ceci pouvant engendrer une grande perte de temps.*

4.1.CALL (APPEL)

Chaque fonction logique activée dispose d'une section appelée "CALL", dans laquelle il faut définir l'objet entrant qui déclenchera l'exécution de la fonction en question.

Il est possible de sélectionner jusqu'à 8 objets déclencheurs pour une même fonction.

***Note :** Pour qu'une fonction s'exécute, il faudra qu'au moins un des objets déclencheurs, sélectionné dans cette section, s'actualise. Ce n'est pas obligatoire d'utiliser les objets déclencheurs de la fonction dans les opérations de celle-ci.*

4.2.OPÉRATIONS

Dans cette section sont définies les opérations que compose la fonction. Dans chaque fonction il est possible d'utiliser jusqu'à 4 opérations de calcul différents.

- **Opération :** Pour utiliser une opération, il est tout d'abord nécessaire de l'**activer**.
- **Type :** Il existe 4 types d'opérations distinctes :
 - **Logic (1 bit) :** Ce type d'opération travaille avec des valeurs de 1 bit. Il est possible de choisir entre les différentes opérations logiques suivantes

ID, AND, OR, XOR, NAND, NOR et NXOR. Chacune d'elle travaille avec deux opérandes (sauf **ID** et **NOT**, qui travaille seulement avec une seule). Les opérandes peuvent être choisis entre les **16 objets de 1 bit** et les **16 variables internes de 1 bit**. Le résultat des opérations de type logique est également de 1 bit et peut être gardé dans n'importe laquelle des 16 variables internes de 1 bit.

- **Aritmetic (1 byte / 2 bytes (unsigned integer (entier sans signe)) / 2 bytes (floating point (virgule flottante)))** : Selon le type choisi, ces opérations travaillent avec des valeurs de 1 byte ou de 2 bytes. Il est possible de choisir entre les opérations arithmétiques suivantes : **ID, ADD, SUBTRACT, MULTIPLY, DIVIDE, MAXIMUM, MINIMUM**. Toutes ces opérations travaillent avec 2 opérandes (sauf **ID** qui travaille avec un seul), qui peuvent être choisis entre les objets, les variables où une valeur constante choisie par paramètre. Le résultat de l'opération arithmétique sera de 1 byte, ou de 2 bytes (selon l'opération choisie). Ce résultat peut être gardé dans n'importe laquelle des 8 variables correspondantes (de 1 ou 2 bytes).

***Note :** L'opération arithmétique (2 bytes entier sans signe) fonctionne avec des données dans le rang (0....65535). Les constantes introduites dans le champ paramétrable correspondant utilisent le format 1X (c-à-d valeur 4000 => paramètre 4000).*

***Note I :** L'opération arithmétique (2 bytes avec virgule flottante) fonctionne avec des données dans le rang (0.....1200). Les constantes introduites dans le champ paramétrable correspondant utilisent le format 0.1X (c-à-d valeur 22.5 => paramètre 225).*

***Note II :** Dans les opérations arithmétiques de 2 bytes, si le résultat excède le rang permit, celui-ci se converti en son extrême correspondant (120). Diviser par "0" n'envoi rien au BUS.*

- **Comparaison (1 byte /2 bytes (entier sans signe) / 2 bytes (virgule flottante))** : Ces opérations travaillent avec des valeurs de 1 ou 2 bytes, selon le type choisi. Il est possible de choisir entre les différentes opérations de comparaison suivantes : **HIGHER, HIGHER o EQUAL, LOWER, LOWER o EQUAL, EQUAL, UNEQUAL**. Toutes celles-ci travaillent avec deux opérandes qui peuvent être choisis entre les objets, les variables ou une valeur constante choisie par paramètre. Le résultat de l'opération est du type 1 bit ("1" si l'opération est vérifié et "0" sinon). Ce résultat peut être sauvegardé dans n'importe laquelle des 16 variables de 1 bit disponible.
- **Conversion (1 bit / 1 byte / 2 bytes (entier sans signe) / 2 bytes (virgule flottante))** : Permet la conversion entre les différents formats des objets de communication.

- **Operation result (Résultat de l'opération)** : Défini la variable dans laquelle sera conservé le résultat de l'opération.

4.2.1. DESCRIPTION DES FONCTIONS – CONVERSION

A continuation sont répertoriées des informations relatives aux fonctions de conversion de l'ACTinBOX CLASSIC-HYBRID :

- **Conversion** (1 bit => 1 byte)

1bit	1byte
0	00000000
1	00000001

- **Conversion** (1 bit => 2 bytes entier sans signe)

1bit	2bytes entier sans signe
0	00000000 00000000
1	00000000 00000001

- **Conversion** (1 bit => 2 bytes virgule flottante)

1bit	2 bytes virgule flottante
0	0
1	0,1

- **Conversion** (1 byte => 1 bit)

1byte	1bit
0	0
1..255	1

- **Conversion** (1 byte => 2 bytes entier sans signe)

1byte	2bytes
\$00	\$00 00
\$01	\$00 01
...	...

\$FF	\$00 FF
------	---------

- **Conversion** (1 byte => 2 bytes virgule flottante)

1byte	2 bytes virgule flottante
0	0
1	0.1
255	25.5

Note : la limite de la conversion est de 25.5

- **Conversion** (2 bytes entier sans signe => 1 bit)

2bytes entier sans signe	1bit
0	0
1..65535	1

- **Conversion** (2 bytes entier sans signe => 1 byte)

2bytes entier sans signe	1byte
\$00 00	\$00
\$00 01	\$01
...	...
\$00 FF	\$FF
> \$00 FF	\$FF

- **Conversion** (2 bytes entier sans signe => 2 bytes virgule flottante)

2bytes entier sans signe	2 bytes virgule flottante
0	0
1	0.1
...	...
1200	120
>1200	120

- **Conversion** (2 bytes virgule flottante => 1 bit)

2 bytes virgule flottante	1bit
0	0
0,1.....120	1

- **Conversion** (2 bytes virgule flottante => 1 byte)

2 bytes virgule flottante	1byte
0	0
0,1... 25,5 .	1..255
> 25,5	255

- **Conversion** (2 bytes virgule flottante => 2 bytes entier sans signe)

2 bytes virgule flottante	2bytes entier sans signe
0	0
0.1	1
...	
120	1200
>120	1200

4.3.RESULT (RÉSULTAT)

Cette section détermine où sera conservé et que faire avec le résultat des opérations de la section antérieure.

- **Type** : Tout d'abord est déterminé le type de résultat, 1 bit, 1 byte ou 2 bytes (entier sans signe) / (virgule flottante)
- **Value (valeur)** : Indiquer quelle variable, provenant des opérations antérieures, enverra son résultat sur l'objet de communication "fonction X RESULT".

***Note :** Il faut se souvenir que les différentes variables sauvegardées sont des variables partagées pour toutes les fonctions logiques possibles, ceci signifie qu'il est préférable qu'une variable utilisée pour gardée le résultat partiel/total d'une opération/fonction ne soit pas réutilisée pour garder le résultat d'une autre fonction.*

- **Send (envoi) :** Défini sous quelles conditions est envoyées le résultat sur le BUS.
 - **Result is diferent from last send (résultat final différent de l'antérieur) :** Le résultat sera envoyé chaque fois que le résultat final provenant des opérations, défini dans les sections antérieures, changera d'état ou de valeur.
 - **Whenever function is execute (quand la fonction est exécutée) :** A chaque fois que la fonction sera exécutée, le résultat sera envoyé.

***Note :** Ce paramètre est lié avec la section **CALL** (voir chapitre 4.1-CALL) : en effet le résultat sera envoyé à chaque fois que s'exécutera la fonction, mais la fonction sera exécutée seulement si au moins un des objets activés, dans la section **CALL**, s'actualise.*

- **Periodical sending (envoi périodique) :** Cette fonction permet d'envoyer le résultat de façon répétitif toutes les Xs (temps déterminé par le paramètre **Cycle Time**).
- **Restriction :** Pour les **fonctions avec un résultat du type 1 bit**, il est possible de limiter l'envoi sur le BUS du résultat de la fonction à une des valeurs possibles ("0" ou "1") (valeur qui sera choisi comme valeur non restreinte). **Pour les fonctions avec un résultat du type 1 byte et 2 bytes**, l'envoi peut être limité aux paramètres suivant:
 - Value equal to reference one (valeur égal à la valeur de référence).
 - Value not equal to reference one (valeur différente à la valeur de référence).
 - Value lower than reference one (valeur inférieure à la valeur de référence).
 - Value higher than reference one (valeur supérieurs à la valeur de référence).
 - **Reference value (valeur de référence) :** Pour un **résultat de type 1 byte** , il est possible d'indiquer une valeur de référence comprise entre **[0.....255]**. Pour un **résultat de type 2 bytes**, il est possible d'indiquer une valeur de référence comprise entre **[0.....65535]**.
- **Delay (retard) :** Il est possible d'établir un temps d'attente entre la réception du résultat et son envoi sur le BUS. Si aucun retard n'est nécessaire choisir la valeur "0"

***Note:** les liens internes qui existaient dans l'ACTinBOX Classic ont été supprimés car le nouveau firmware 5.0 supporte le Full Duplex.*

5. OBJETS DE COMMUNICATION

Les objets de communication des fonctions logiques sont de deux types :

- **Data (Données)** : Ce sont les données provenant du BUS avec lesquelles travaillent les fonctions.
- **Result (Résultats)** : Ce sont les résultats des fonctions logiques (un pour chacune d'elles). Selon la taille de ces objets, elles se divisent en 3 types : 1 bit, 1 byte et 2 byte.

5.1.NOMENCLATURE :

- **Objet de type : Données**

[LF]("**Taille**") **Data Entry "X"** où "taille" peut être de 1 bit, 1 byte ou 2 bytes ; et "X" est le numéro de la donnée (c'est-à-dire, entre 1 et 16 pour les données de 1 bit , et entre 1 et 8 pour ceux de 1 et 2 bytes.

- **Objet de type : Résultat**

[LF]**function "X" RESULT ("**taille**")** où "taille" peut être du type 1 bit, 1 byte ou 2 bytes (dépend du type de donnée qu'aura la fonction comme résultat), et "X" est le numéro de la fonction (c'est-à-dire entre 1 et 5).

- **Variables internes**

b1,....., b16 (pour ceux du type de 1 bit)

n1,....., n8 (pour ceux du type de 1 byte)

x1,....., x8 (pour ceux du type de 2 bytes)



Devenez membre !

<http://zennio.zendesk.com>

SUPPORT TECHNIQUE