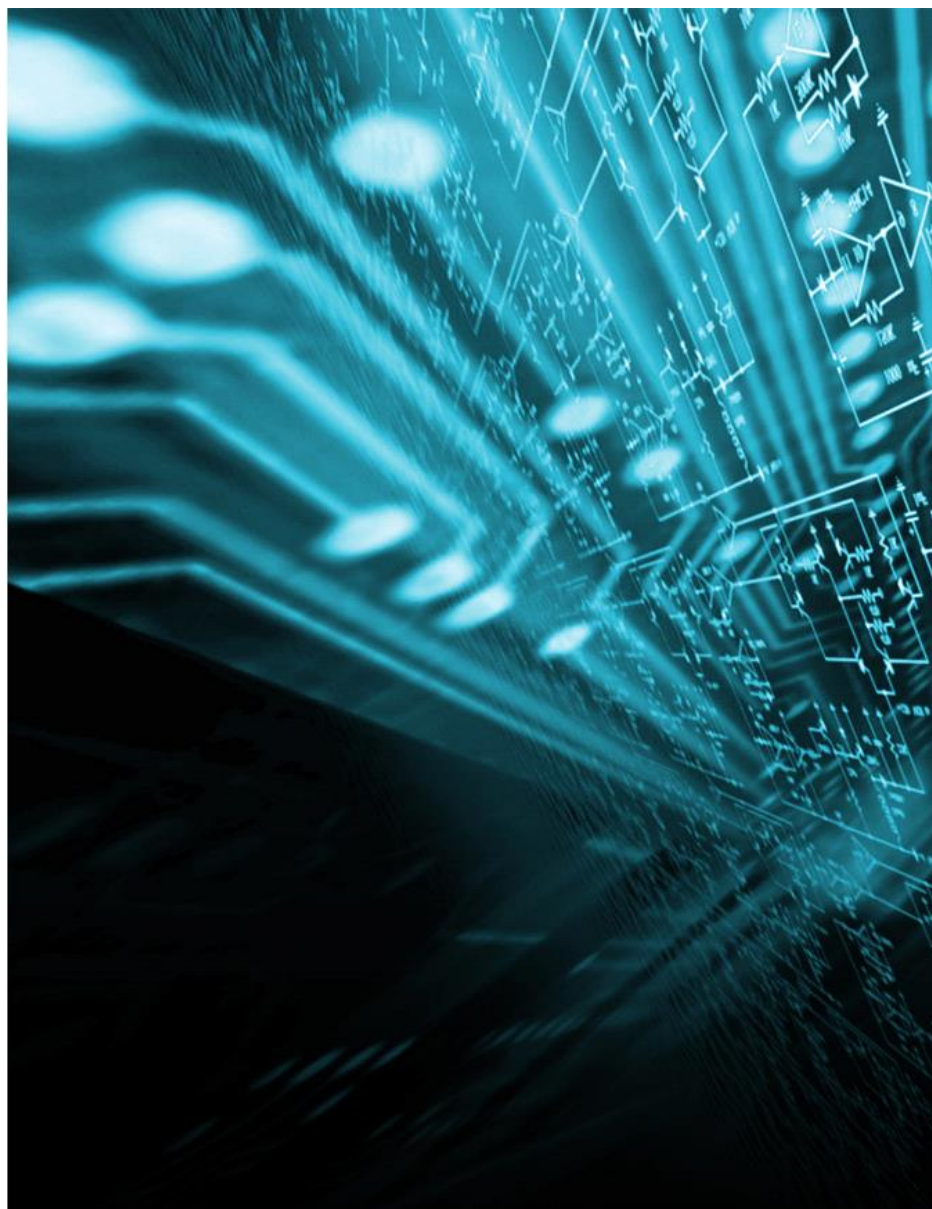


# Manuel du Produit



# WBL

## Module Logique

### WEB-BASED

L'accès à l'interface du produit ne nécessite aucune installation. Elle se fait via un navigateur web (IE, Chrome, Firefox, Safari).  
NB : connexion à internet pas requise.

### VISUEL

Visuel et intuitif, WBL facilite grandement la mise en œuvre d'automatismes complexes sur votre installation domotique.

### KNX

Les fonctionnalités d'écriture et de lecture sur le bus KNX vous permettent de traiter les données captées et de communiquer simplement avec les actionneurs.

### GPIO (General Purpose Input/Output)

La lecture et l'écriture des ports d'entrée/sortie permettent d'interagir avec des circuits électroniques.

### ALERTES

Le déclenchement et l'envoi automatique de SMS et/ou MAIL permet de vous informer rapidement en cas de situations urgentes.

## Table des Matières

1	Installation .....	4
1.1	IP Locale .....	4
1.2	Activation .....	4
1.3	Authentification .....	5
2	Circuits WBL .....	6
2.1	Portes .....	6
2.1.1	Portes Logiques Standards .....	8
2.1.2	Portes Logiques Etendues .....	9
2.1.3	Déclencheurs .....	10
2.1.4	Actionneurs .....	11
2.1.5	Contrôleurs de Flux .....	13
2.1.6	Autres Portes .....	15
2.2	Liens .....	17
2.2.1	Lien Mémoire .....	17
2.2.2	Lien Mémoire Passif .....	17
2.2.3	Lien Consommable .....	17
2.2.4	Synchronisation .....	17
2.3	Valeurs .....	18
2.3.1	Nombres .....	18
2.3.2	Chaînes .....	19
2.3.3	Tableaux .....	19
2.3.4	Inversion des Valeurs .....	19
2.4	Adresses .....	20
2.4.1	Adresses de Groupe KNX .....	20
2.4.2	Adresses Physiques KNX .....	21
2.4.3	Adresses GPIO .....	22
2.4.4	Adresses Virtuelles .....	22
2.4.5	Alias .....	23
2.5	Fonctionnalités Avancées .....	24
2.5.1	Mode Avancé .....	24
2.5.2	Planification de Tâches .....	27

2.5.3	Expressions Avancées .....	28
2.5.4	Porte Programmable (JavaScript) .....	29
3	Interface.....	35
3.1	Panel de Sélection.....	35
3.2	Layout.....	36
3.3	Edition .....	37
3.4	Débug.....	38
3.5	Statut.....	39

# WBL – Module Logique

Le module logique, par l'élaboration de circuits logiques simplifiés, est un moyen puissant d'exploiter les possibilités d'une installation domotique. Concrètement, un circuit logique permet de lier et d'enchaîner plusieurs opérations qui peuvent être conditionnées par des informations reçues des capteurs, mais aussi par vos propres paramètres. A l'aide de notre interface web, il est facile de construire rapidement des circuits interactifs exécutants des automatismes complexes. L'exemple introductif (Figure 1) illustre comment une tâche relativement difficile à coder est simplement réalisée et ne nécessite pas de connaissances en programmation informatique.

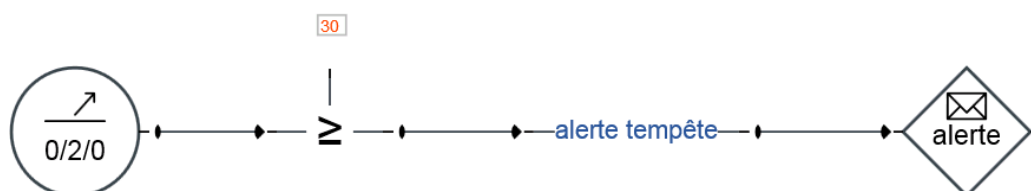


Figure 1 Une alerte SMS contenant le message « alerte tempête » est envoyée automatiquement lorsque la vitesse du vent mesurée par l'anémomètre circulant sur l'adresse de groupe KNX 0/2/0 dépasse les 30 m/s

Principales fonctionnalités offertes par le module logique :

- **Edition/Import/Export de circuits logiques** (automatismes)
- **GPIO** (General Purpose Input/Output) supporté selon le hardware (Raspberry)
  - Lecture des bits reçus en entrée (Input)
  - Ecriture et lecture des bits envoyés en sortie (Output)
- **KNX**
  - Import des adresses KNX configurées depuis **ETS**
  - Connexion aux bus KNX possible via des **passerelles KNX** de type **IP ou USB**
  - **Multi-passerelles** : possibilité d'être connecté et d'interagir avec plusieurs bus KNX simultanément
  - Lecture des télégrammes de communication passant sur le bus KNX
  - Détection et ajout automatique des adresses communiquant sur le bus KNX
  - Ecriture et envoi de requêtes de lecture sur les adresses de groupe (p.ex. 1/2/3)
  - Réinitialisation d'un participant via son adresse physique (p.ex. 1.1.1)

Ce document est divisé en trois chapitres :

- Chapitre 1: instructions pour l'installation du module logique
- Chapitre 2: présentation complète des circuits et de leur fonctionnement
- Chapitre 3: description de l'interface web du module logique

# 1 INSTALLATION

## 1.1 IP LOCALE

L'installation de l'appareil se fait comme pour un routeur. Une fois démarré, l'adresse IP de l'appareil par défaut est 192.168.1.99. Une fois l'appareil connecté au réseau, on peut accéder à son interface via un navigateur web (IE/Firefox/Chrome/Safari/..) en pointant l'adresse suivante : <http://192.168.1.99>

## 1.2 ACTIVATION

Pour pouvoir utiliser le module, il faut l'activer (Figure 2). Son activation « en ligne » se fait à l'aide du « code client » fourni avec le produit. Il est aussi possible d'activer l'appareil hors ligne à l'aide de la « License matérielle ». L'activation « hors ligne » ne donne pas accès aux mises à jour et correctifs.

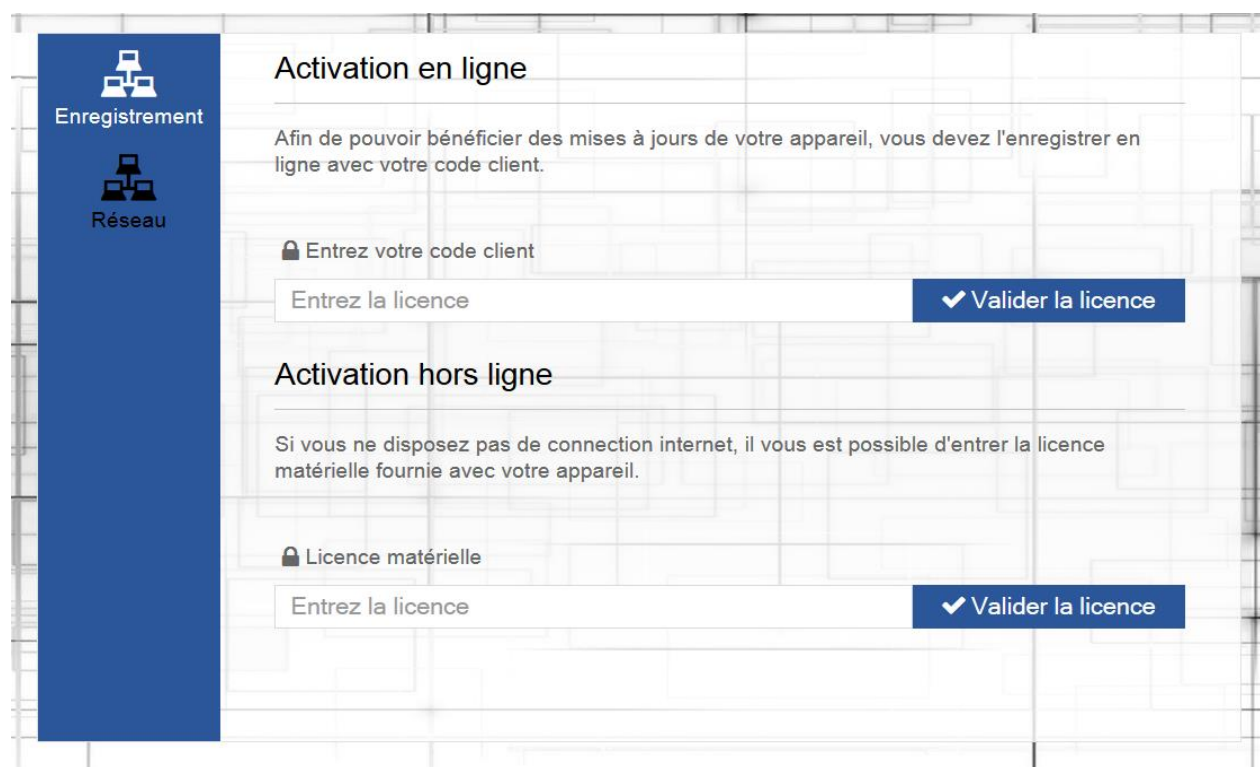


Figure 2 L'activation du produit peut se faire en ligne ou hors-ligne. Pour pouvoir bénéficier des correctifs et mises à jour, l'activation en ligne est nécessaire.

### 1.3 AUTHENTIFICATION

L'accès à l'appareil est protégé par un nom d'utilisateur et un mot de passe. A la première utilisation le compte administrateur par défaut est : utilisateur « admin », mot de passe « admin » (Figure 3). Par la suite l'administrateur pourra gérer depuis l'interface les comptes utilisateurs comme il le souhaite.

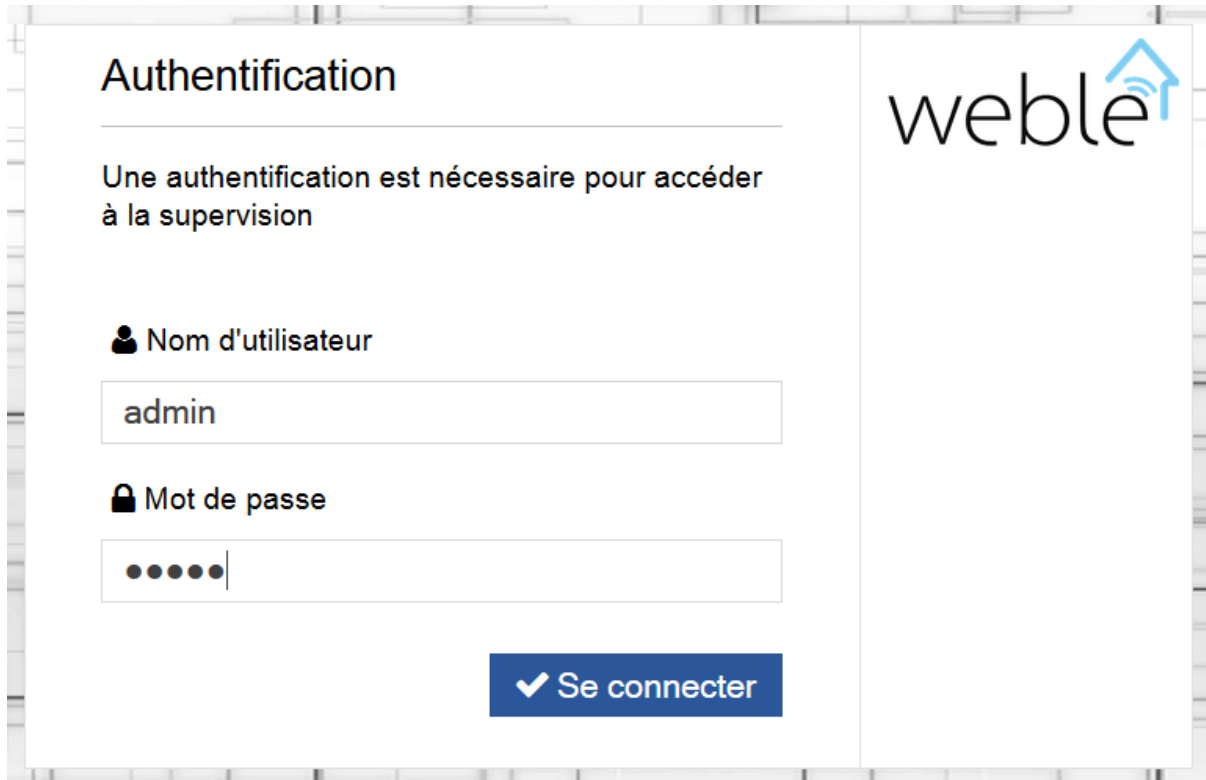


Figure 3 Une fois le produit activé, à l'installation le compte par défaut est utilisateur: "admin", mot de passe: "admin"

## 2 CIRCUITS WBL

---

Les circuits WBL sont une adaptation des circuits logiques standards. Ils reprennent et étendent de façon naturelle leurs fonctionnalités par des mécanismes simples et efficaces.

Les circuits se composent de portes (Section 2.1) reliées entre elles par des liens directionnels (Section 2.2) transportant des valeurs (Section 2.3) et propageant ainsi l'exécution.

L'interaction avec les capteurs et actionneurs du bus KNX ou les entrées et sorties digitales (GPIO) se fait par un système d'adresses (Section 2.4).

Les cas d'utilisation avancés sont abordés à la dernière section de ce chapitre (Section 2.5). Ils nécessitent un peu plus de connaissances, mais leur maîtrise permet de répondre à toutes les situations.

### 2.1 PORTES

Les circuits comportent une vingtaine de portes différentes (Table 1), et chacune d'elle a une série de paramètres configurables. Cette souplesse permet de concevoir facilement des automatismes complexes.

Les portes sont subdivisées en catégories :

- **Les portes logiques :**
  - **Les portes logiques standards** (Section 2.1.1) : fonctions logiques de base tel que *OU/ET*.
  - **Les portes logiques étendues** (Section 2.1.2) : fonctions utiles à la manipulation de valeurs logiques (**1/0**) et plus généralement de valeurs numériques (p. ex. **1.233**)
- **Les déclencheurs** (Section 2.1.3) : les déclencheurs attendent qu'un événement se produise pour déclencher l'exécution d'actions. Ils permettent entre autres de réaliser des tâches planifiées, ou de capter des télégrammes passant sur le bus KNX.
- **Les actionneurs** (Section 2.1.4) : les actionneurs permettent d'effectuer des actions concrètes tel que l'écriture sur le bus KNX ou l'envoi d'alertes SMS et/ou MAIL.
- **Les contrôleurs de flux** (Section 2.1.5) : les contrôleurs de flux jouent un rôle important d'aiguillage permettant de contrôler et synchroniser les flux d'exécutions sur le circuit.
- **Autres portes** (Section 2.1.6) : cette dernière catégorie comporte des annotations (effet purement visuel et informatif) et des portes plus avancées tel que la porte programmable (Section 2.5.4).

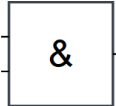
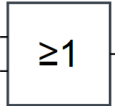
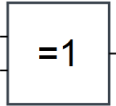
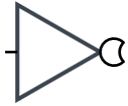

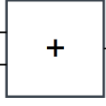
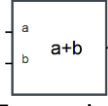

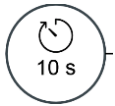

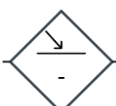
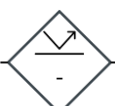





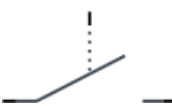


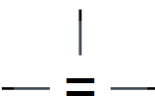





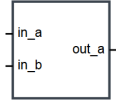
PORTES LOGIQUES	 ET	 OU	 OU Exclusif	 Inverse
	 Constante	 Calcul	 Expression	
DÉCLENCHEURS	 Adresse	 Horloge	 Réception	
ACTIONNEURS	 Ecriture	 Lecture	 Réinitialisation	 Alerte
	 Emission			
CONTRÔLEURS DE FLUX	 Jonction	 Disjonction	 Commutation	 Sélection Sorties
	 Sélection Entrées	 Filtre	 Délai	 Mémoire Tampon
AUTRES PORTES	<b>Exemple d'Annotation</b> Annotation	 Groupe	 Fusion	 Scission
	 Programmable			

Table 1 Aperçu par catégories de toutes les portes disponibles.



### 2.1.1 Portes Logiques Standards

Les portes logiques standards de base sont la représentation graphique des fonctions logiques élémentaires. Réaliser ces opérations logiques permet en pratique par exemple de tester si certaines conditions sont remplies ou non avant d'enclencher une action.

Traditionnellement en algèbre de Boole, une donnée, qu'elle soit en entrée ou en sortie, n'a que deux valeurs possibles : **1/0**. La Table 2 présente les portes ET, OU, et OU exclusif ainsi que leur table de vérité décrivant la valeur de la sortie en fonction des valeurs d'entrée.

Comme on le voit sur la Figure 4, on peut inverser les entrées et sorties de ces portes et aussi y ajouter des entrées supplémentaires.

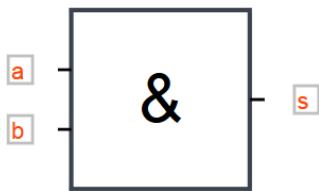
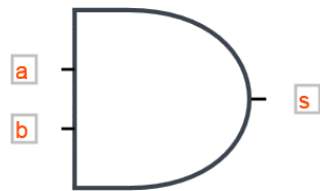
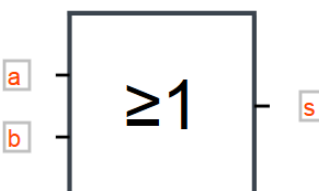
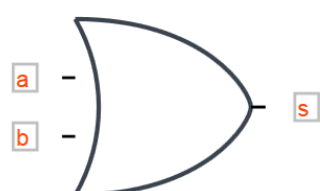
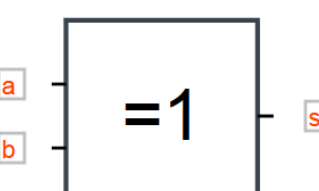

FONCTION LOGIQUE	SYMBOLE EUROPÉEN	SYMBOLE AMÉRICAIN	TABLE DE VÉRITÉ															
<b>ET</b> ( $n \rightarrow 1$ ) Envoie <b>1</b> sur la sortie seulement si toutes les entrées sont à 1			<table><tr><th>a</th><th>b</th><th>s</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	s	0	0	0	0	1	0	1	0	0	1	1	1
a	b	s																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
<b>OU</b> ( $n \rightarrow 1$ ) Envoie <b>1</b> sur la sortie seulement si au moins une des entrées est définie à <b>1</b>			<table><tr><th>a</th><th>b</th><th>s</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	a	b	s	0	0	0	0	1	1	1	0	1	1	1	1
a	b	s																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
<b>OU Exclusif</b> ( $n \rightarrow 1$ ) Envoie <b>1</b> sur la sortie seulement si exactement une entrée est définie à <b>1</b>			<table><tr><th>a</th><th>b</th><th>s</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	a	b	s	0	0	0	0	1	1	1	0	1	1	1	0
a	b	s																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Table 2 Définition des portes logiques standards. Les formats européens et américains sont tous deux supportés par l'interface.

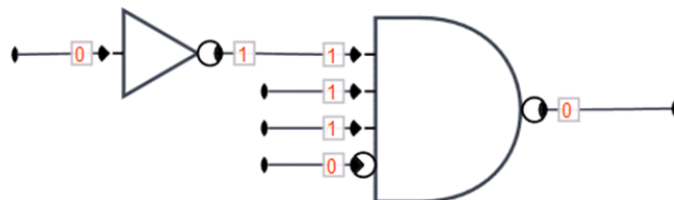
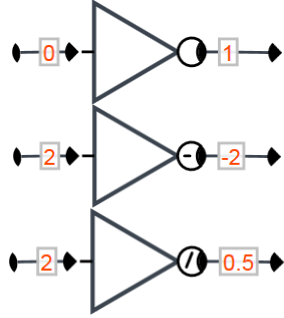
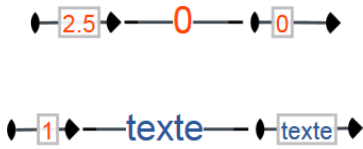
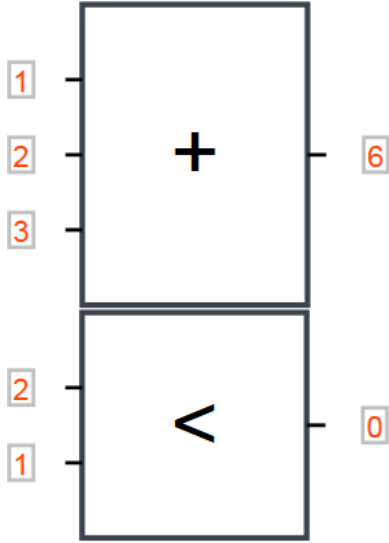


Figure 4 Illustre comment il est possible d'inverser les entrées et les sorties des portes logiques, soit par l'ajout d'un inverseur, soit directement sur les entrées et sorties de la porte. Notez aussi qu'ici deux entrées supplémentaires ont été ajoutées à la porte ET.

### 2.1.2 Portes Logiques Etendues

Les circuits logiques ne se limitent pas seulement au traitement de valeurs booléennes. Les liens reliant les différentes portes du circuit peuvent transporter d'autres types de valeurs, tel que des nombres (entiers ou rationnels), ou des textes. Les portes logiques étendues, décrites dans la Table 3, répondent à ces cas où les portes logiques standards ne sont pas adaptées.

DESCRIPTION DE LA PORTE LOGIQUE	SYMBOLE
<p><b>INVERSE</b> (<math>1 \rightarrow 1</math>)</p> <p>La porte inverse permet d'inverser la valeur circulant sur le lien. Il existe 3 types d'inversion possibles. L'inverse binaire est <b>0</b> si l'entrée est différente de <b>0</b>, et un <b>1</b> si l'entrée est égale à <b>0</b>. L'inverse additif équivaut à renvoyer sur la sortie <math>-x</math>, où <math>x</math> est la valeur d'entrée. De façon similaire l'inverse multiplicatif équivaut à renvoyer <math>1/x</math></p>	
<p><b>CONSTANTE</b> (<math>1 \rightarrow 1</math>)</p> <p>La constante écrase et remplace la valeur actuelle circulant sur le lien par une valeur constante prédéfinie dans les paramètres de la porte.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Valeur de la constante:</b> valeur de n'importe quelle type (binaire, numérique, chaîne, ..).</li> </ul>	
<p><b>CALCUL</b> (<math>n \rightarrow 1</math>)</p> <p>La porte calcul peut effectuer une somme (+), un produit (<math>\times</math>), une moyenne arithmétique (<math>\mu</math>), ou encore trouver la valeur minimum (<b>min</b>) et maximum (<b>max</b>) des entrées.</p> <p>Les opérateurs de comparaisons <math>&lt;</math>, <math>&gt;</math>, <math>\leq</math>, <math>\geq</math>, <math>=</math>, <math>\neq</math> sont aussi utilisables. Dans le cas d'une comparaison, la sortie retourne le résultat sous forme binaire : <b>1</b> (vrai) / <b>0</b> (faux).</p> <p>Note : la somme (+) a un double usage car elle permet aussi de concaténer des chaînes de caractères (Figure 5).</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Opération :</b> somme (+) / produit (<math>\times</math>) / moyenne (<math>\mu</math>) / maximum (<b>max</b>) / minimum (<b>min</b>) / plus petit (<math>&lt;</math>) / plus grand (<math>&gt;</math>) / plus petit ou égale (<math>\leq</math>) / plus grand ou égal (<math>\geq</math>) / égal (=) / inégal (<math>\neq</math>)</li> </ul>	

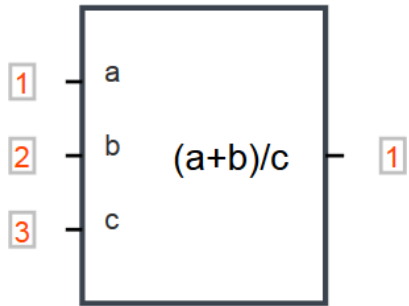
DESCRIPTION DE LA PORTE LOGIQUE	SYMBOLE
<p><b>EXPRESSION</b> (<math>n \rightarrow 1</math>)</p> <p>La porte expression est moins restreinte que la porte calcul. Elle rend possible l'écriture d'expressions mathématiques. On peut attribuer à chacun des inputs un identifiant, et ainsi y référer dans le calcul. Cette porte ne se limite pas aux formules mathématiques : elle permet de réaliser des opérations variées (voir Section 2.5.2).</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Expression</b> : expression mathématique ici <math>(a+b)/c</math></li> </ul>	

Table 3 Description des portes logiques étendues.

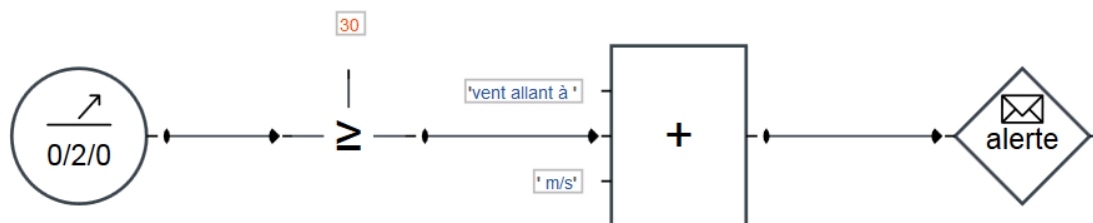
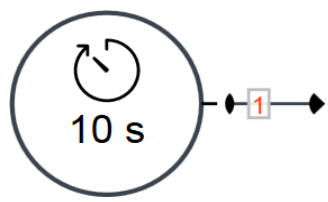


Figure 5 L'opérateur « + » de la porte calcul permet de concaténer les chaînes de caractères et ainsi d'envoyer par SMS/MAIL le contenu "vent allant à 31 m/s" lorsque la vitesse du vent est mesurée à 31 m/s

### 2.1.3 Déclencheurs

Les déclencheurs permettent de réagir à des événements externes (par exemple à des télégrammes circulant sur le bus KNX). Lorsque l'évènement en question se produit, alors la sortie du déclencheur est exécutée. Par convention tous les déclencheurs sont symbolisés par un cercle. La Table 4 décrit les déclencheurs.

DESCRIPTION DU DÉCLENCEUR	SYMBOLE
<p><b>HORLOGE</b> (<math>0 \rightarrow 1</math>)</p> <p>L'Horloge enclenche une nouvelle exécution en envoyant un <b>1</b> sur le lien tous les x intervalles de temps.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Répéter</b> : oui / non</li> <li><b>Unité de temps</b> : seconde / minute / heure / jour</li> <li><b>Délai</b> : nombre d'unités de temps (p. ex 10 secondes)</li> </ul>	

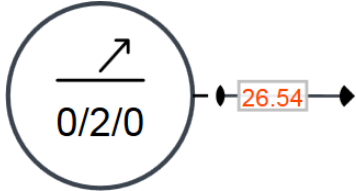
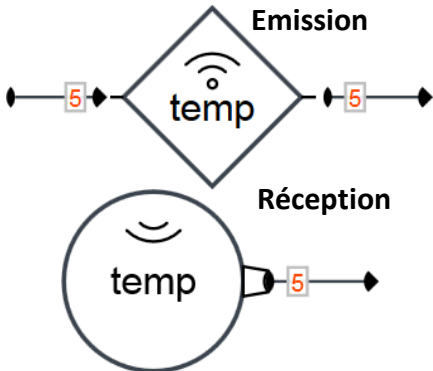
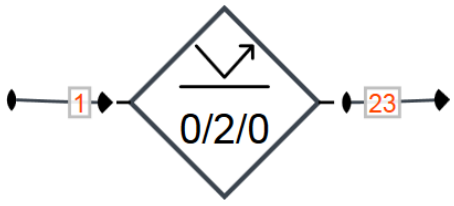
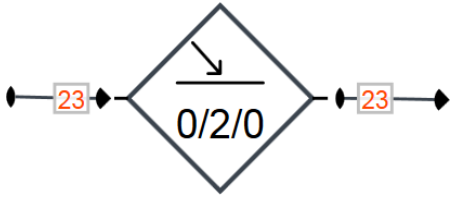
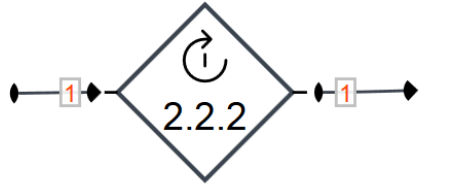

DESCRIPTION DU DÉCLENCHEUR	SYMBOLE
<p><b>ADRESSE</b> (0 → 1)</p> <p>Cette porte écoute sur le bus les télégrammes qui concernent l'adresse de groupe KNX paramétrée. Il envoie à la sortie les dernières valeurs lues.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>Envoyer valeur au démarrage:</b> oui/non</li> <li>• <b>Mode de lecture:</b> chaque valeur / chaque changement de valeur</li> <li>• <b>Adresse de lecture:</b> adresse de groupe KNX</li> </ul>	
<p><b>RECEPTION</b> (0 → 1)</p> <p>Le couplage des portes « émissions » et « réceptions » permet de créer des liens sans fil. La porte réception écoute les valeurs transmises sur le canal donné par l'identifiant paramétré. Une fois la valeur reçue sur le canal, celle-ci est reportée sur la sortie de la porte réception.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>Propagation:</b> Interne / Externe. Définit si on capte les valeurs transmises par tous les circuits (Externe) ou seulement celles transmises à partir du même circuit (Interne)</li> <li>• <b>Canal:</b> clé textuelle identifiant le canal de réception (ici "temp")</li> </ul>	

Table 4 Description des différents déclencheurs.

#### 2.1.4 Actionneurs

En pratique les actionneurs sont utilisés pour lire ou écrire sur le bus KNX, faire un reset d'un actionneur/capteur KNX, ou envoyer des alertes par SMS/MAIL. Par convention tous les actionneurs sont symbolisés par un losange. La Table 5 décrit les principaux actionneurs.

DESCRIPTION DE L'ACTIONNEUR	SYMBOLE
<p><b>LECTURE</b> (1 → 1)</p> <p>Lorsque la porte de lecture reçoit une nouvelle impulsion en entrée, il lit la valeur sur l'adresse de groupe KNX et propage celle-ci sur la sortie.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>Mode de lecture</b> : requête de lecture sur le bus / dernière valeur connue</li> <li>• <b>Adresse de lecture</b> : adresse de groupe KNX</li> </ul>	
<p><b>ECRITURE</b> (1 → 1)</p> <p>L'actionneur d'écriture prend en entrée la valeur à écrire sur l'adresse de groupe paramétrée. Si le télégramme a été correctement envoyé sur le bus, la valeur d'écriture est retranscrite sur la sortie.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>Adresse d'écriture</b> : adresse de groupe KNX</li> </ul>	
<p><b>RÉINITIALISATION</b> (1 → 1)</p> <p>Cet actionneur permet d'envoyer un télégramme KNX <i>reset</i> à un participant (adresse physique). C'est utile en cas de dépannage ou de dysfonctionnements d'un capteur/actionneur. Si le capteur/actionneur a pu redémarrer, la valeur d'entrée est reproduite sur la sortie.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>Adresse du participant</b> : adresse de participant KNX</li> </ul>	
<p><b>ALERTE</b> (1 → 1)</p> <p>L'alerte permet d'envoyer des MAIL et/ou SMS afin d'alerter un responsable en cas d'urgences ou de diffuser en externe des informations relatives à l'installation domotique. Elle prend en entrée le texte à envoyer.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>Destinataires</b> : numéro de téléphone / adresse(s) MAIL</li> <li>• <b>CLIP / Objet</b> : intitulé du SMS / Objet du MAIL</li> </ul>	

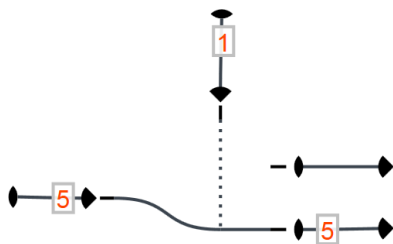
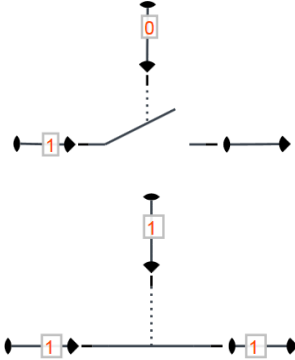
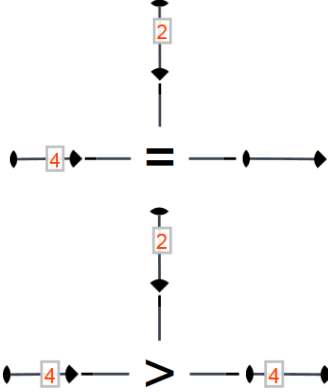

DESCRIPTION DE L'ACTIONNEUR	SYMBOLE
<p><b>EMISSION</b> (<math>1 \rightarrow 1</math>)</p> <p>Le couplage des portes « émissions » et « réceptions » permet de créer des liens sans fils. Lorsqu'il reçoit une valeur en entrée, l'actionneur émission la retransmet sur son canal.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>Propagation:</b> Interne/Externe. Définit si la retransmission se restreint à son propre circuit (Interne) ou si elle s'étend à tous les autres circuits (Externe).</li> <li>• <b>Identifiant l'évènement :</b> clé textuelle ici "temp"</li> </ul>	

Table 5 Description des différents actionneurs.

### 2.1.5 Contrôleurs de Flux

Les portes contrôlant le flux d'exécution sont au cœur des circuits. Elles jouent le rôle indispensable d'aiguillage et influent la façon dont les valeurs sont traitées et propagées dans le circuit. La Table 6 présente les contrôleurs de flux et les figures ci-dessous (Figure 6, Figure 7, Figure 8) donnent des exemples concrets de leur utilisation.

DESCRIPTION DU CONTRÔLEUR DE FLUX	SYMBOLE
<p><b>JONCTION</b> (<math>n \rightarrow 1</math>)</p> <p>La jonction lie plusieurs entrées distinctes sur la même sortie. Les valeurs reçues en entrée sont répercutées instantanément sur la sortie.</p>	
<p><b>DISJONCTION</b> (<math>1 \rightarrow n</math>)</p> <p>Lorsqu'une valeur est reçue en entrée, celle-ci est répliquée et envoyée en parallèle sur les différentes sorties.</p>	
<p><b>SÉLECTION ENTRÉES</b> (<math>n \rightarrow 1</math>)</p> <p>L'entrée du haut de l'aiguillage permet de sélectionner l'entrée opérationnelle. Les entrées de gauche sont indexées à partir de 0. Par exemple, si l'aiguillage est configuré sur <b>1</b>, alors la deuxième entrée est sélectionnée.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>• <b>synchronisation :</b> oui/non. Définit si l'ensemble des entrées doivent avoir une valeur avant de pouvoir poursuivre l'exécution (voir Section 2.2.4).</li> </ul>	

DESCRIPTION DU CONTRÔLEUR DE FLUX	SYMBOLE
<p><b>SÉLECTION SORTIES</b> (<math>1 \rightarrow n</math>)</p> <p>L'entrée d'aiguillage du bas permet de sélectionner la sortie choisie. Par exemple, si l'aiguillage est configuré sur <b>1</b>, alors la valeur de la deuxième entrée (en comptant du haut vers le bas à partir de <b>0</b>) est redirigée sur la sortie.</p>	
<p><b>COMMUTATION</b> (<math>1 \rightarrow 1</math>)</p> <p>Si l'entrée du poussoir est sur <b>1</b>, alors l'exécution de l'entrée standard (à gauche) peut se propager sur la sortie. Au contraire, si le poussoir a la valeur <b>0</b>, l'exécution est stoppée.</p>	
<p><b>FILTRE</b> (<math>1 \rightarrow 1</math>)</p> <p>L'input du haut est comparé à l'entrée standard (gauche), et si la condition du filtre n'est pas remplie, alors l'exécution est stoppée.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Comparateur:</b> plus petit (<math>&lt;</math>) / plus grand (<math>&gt;</math>) / plus petit ou égale (<math>\leq</math>) / plus grand ou égal (<math>\geq</math>) / égal (<math>=</math>) / inégal (<math>\neq</math>)</li> </ul>	
<p><b>DÉLAI</b> (<math>1 \rightarrow 1</math>)</p> <p>La porte délai retarde d'un certain laps de temps la propagation de l'exécution. Si la porte est réactivée par une nouvelle valeur avant que l'ancien délai n'ait échu, alors différentes réactions sont possibles. Celles-ci sont listées dans le paramètre traitant les entrées concomitantes.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Unité du délai:</b> milliseconde/seconde/minute/heure/jour</li> <li><b>Délai:</b> nombre d'unités de temps</li> <li><b>Entrées concomitantes:</b> Ignore la nouvelle entrée/Réinitialise le délai/Cumule les délais/Exécute les délais en parallèle</li> </ul>	





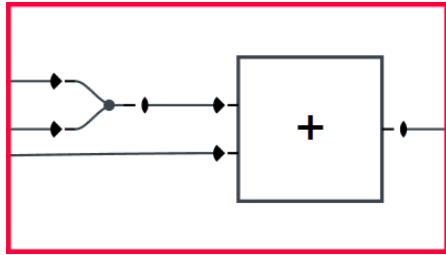
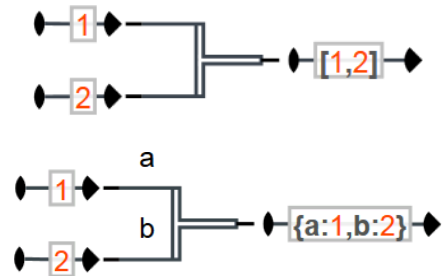
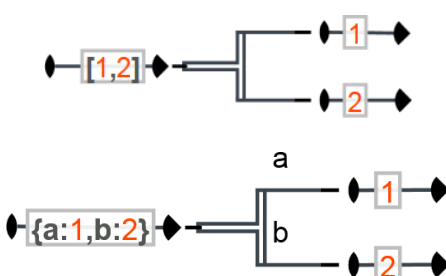
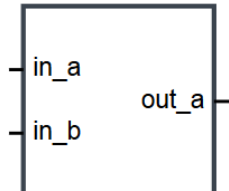
DESCRIPTION DE LA PORTE	SYMBOLE
<p><b>ANNOTATION</b></p> <p>Permet d'ajouter du texte purement informatif dans les circuits logiques.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>Couleur, Police d'écriture, Interligne, Gras, Souligné.</li> </ul>	<p>Exemple d'annotation</p>
<p><b>GROUPE</b></p> <p>Permet de grouper visuellement différentes portions du circuit logique.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li>Couleur bordure, Epaisseur bordure</li> </ul>	
<p><b>FUSION</b> (<math>1 \rightarrow 1</math>)</p> <p>Fusionne les valeurs reçues en entrée dans un tableau (voir Section 2.3.3).</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Associatif</b> : oui/non. Décide si un tableau associatif est créé à la sortie de la porte. Si c'est le cas, alors les entrées sont identifiées par des clés paramétrables.</li> <li><b>Synchronisation</b> : oui/non. Définit si l'ensemble des entrées doivent être définies avant de pouvoir poursuivre l'exécution (voir Section 2.2.4).</li> </ul>	
<p><b>SCISSION</b> (<math>1 \rightarrow 1</math>)</p> <p>Sépare les valeurs d'un tableau (voir Section 2.3.3) et les envoie sur les différentes sorties.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><b>Associatif</b> : oui/non. Définit si un tableau associatif est attendu. Si c'est le cas alors des clés sont introduites afin d'accéder aux éléments du tableau associatif.</li> </ul>	
<p><b>PROGRAMMABLE</b> (<math>n \rightarrow m</math>)</p> <p>La porte programmable (Section 2.5.4) est une porte avancée qui permet d'implémenter ses propres fonctionnalités logiques en langage de programmation JavaScript. Ses entrées/sorties et ses paramètres sont entièrement configurables.</p>	

Table 7 Description des portes appartenant à la catégorie « autres portes ».

## 2.2 LIENS

Les liens lient les différentes portes entre elles et propagent ainsi l'exécution dans le circuit.

Il existe trois types de liens :

- Lien mémoire (Section 2.2.1)
- Lien mémoire passif (Section 2.2.2)
- Lien consommable (Section 2.2.3)

Ces types de liens sont utiles à la synchronisation des flux d'exécution (Section 2.2.4)

### 2.2.1 Lien Mémoire



Le lien mémoire conserve toujours sa valeur courante. Lorsqu'il est exécuté, l'ancienne valeur est remplacée par la nouvelle.

### 2.2.2 Lien Mémoire Passif



Le lien mémoire passif est similaire au lien mémoire normal, à l'exception qu'il ne propage l'exécution que si la nouvelle valeur reçue est **différente** de l'ancienne : par exemple passage de 0 → 1 (flux montant) ou de 1 → 0 (flux descendant).

### 2.2.3 Lien Consommable



Les liens reliant les portes peuvent ne contenir aucune valeur. Dans ce cas, l'absence de valeur est dénotée par le symbole  $\emptyset$ . Le lien consommable, à contrario des liens mémoire, perd sa valeur (elle revient à  $\emptyset$ ) lorsque celle-ci est consommée par la porte ciblée.

### 2.2.4 Synchronisation

**Avant de pouvoir s'exécuter, les portes doivent attendre que toutes leurs valeurs d'entrée soient bien définies. Autrement dit, si une des entrées de la porte est égale à  $\emptyset$ , alors le flux d'exécution reste bloqué en attente.**

L'application de cette règle associée aux différents types de liens offre un mécanisme de synchronisation simple et efficace. Par exemple cela permet la réalisation d'un compteur comme illustré sur la Figure 9.

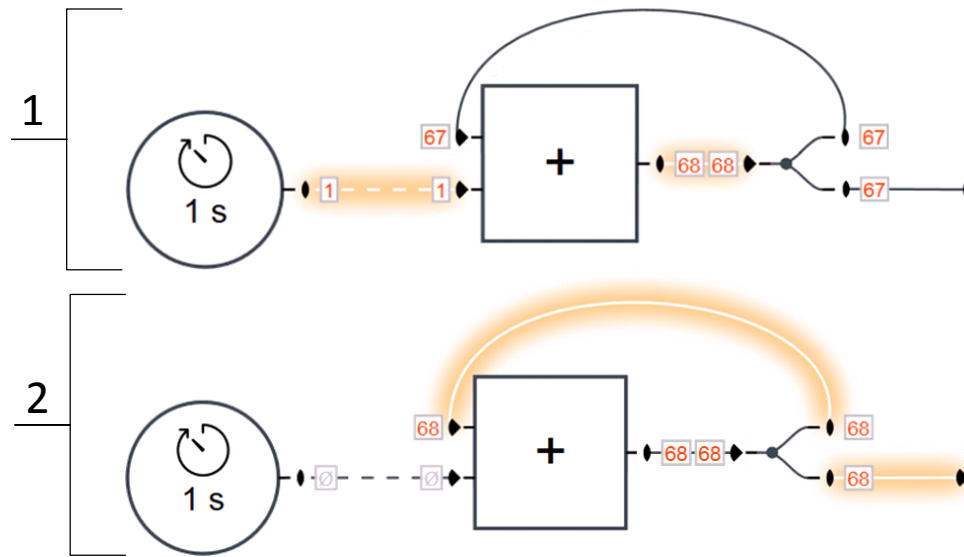


Figure 9 Cette figure décompose l'exécution d'un compteur incrémenté de **1** chaque secondes. Notez que le lien consommable perd sa valeur lorsque l'addition est effectuée. Ceci oblige la porte calcul à attendre la prochaine impulsion avant de pouvoir effectuer à nouveau l'incrémentation du compteur (évite d'avoir une boucle infinie). Le lien mémoire, quant à lui, permet de stocker la valeur courante du compteur (ici **68**).

## 2.3 VALEURS

Les circuits ne sont pas limités aux seules valeurs booléennes (**1/0**). Trois types de valeurs peuvent transiter sur les liens :

- Des nombres : **1.23** (Section 2.3.1)
- Des chaînes (de caractères) : **exemple de chaîne** (Section 2.3.2)
- Des tableaux de valeurs : [**0**, **1**, **exemple de chaîne**] (Section 2.3.3)

De façon similaire aux circuits logiques traditionnels, ces valeurs peuvent être inversées aux entrées et sorties des portes (Section 2.3.4).

### 2.3.1 Nombres

La représentation interne des nombres est toujours à virgule flottantes avec précision double (64 bits) suivant le standard international IEEE 754.

Au niveau de la syntaxe, plusieurs notations différentes sont possibles :

- Nombres à virgules: **0.01**, **10.00**, **-10.505**
- Entiers (base 10) : **0**, **1**, **2**, **-2**, **12456**
- Entiers hexadécimaux (base 16) : **0xFF** (= 255), **0x12** (= 18), **0xA** (= 10), **-0xA** (= -10)
- Entiers octadécimaux (base 8) : **001** (= 1), **00100** (= 64), **0077** (= 63), **-0077** (= -63)
- Notation scientifique: **1e2** (= 100), **-1.2e-2** (= -0.012), **0.0023e+4** (= 23)
- Valeurs spéciales : **PI** (= 3.141...), **E** (= 2.718...), **∞**, **Infinity** (infini), **NaN** (Not a Number)

- Expressions calculatoires:  $(\sin(\pi/4)+0.5)*2$  ( = 2.414...),  $\sqrt{5}+\log(10)$  ( = 4.538...). Fonctions disponibles : [http://www.w3schools.com/js/js\\_math.asp](http://www.w3schools.com/js/js_math.asp)

### 2.3.2 Chaînes

Les chaînes de caractères permettent de former des textes : [Exemple de chaîne](#). Elles peuvent être concaténées entre elles ou avec des nombres afin de former une nouvelle chaîne (Figure 5, Figure 7). Lorsqu'une chaîne de caractère peut être confondue avec un nombre, l'ajout de guillemets (" ou ') autour de celle-ci permet la désambiguïsation :  $\pi \rightarrow "PI"$ ,  $1.23 \rightarrow '1.23'$ ,  $\sqrt{5}+\log(10) \rightarrow '\sqrt{5}+\log(10)'$

Note : des formats particuliers de chaînes font l'objet d'un traitement spécial par certaines portes :

- Les **adresses** d'écriture et de lecture (Section 2.4)
- Les **numéros de téléphone et adresses emails** (interprétés par la porte « alerte »)
- Le système de **dates** pour la planification de tâches (Section 2.5.2)

### 2.3.3 Tableaux

Les tableaux permettent de passer plusieurs valeurs sur le même lien. Deux types de tableaux sont possibles :

- Les **tableaux simples**. Les tableaux simples sont délimités par des accolades carrées [...]. Les valeurs sont séparées par des virgules et chaque valeur est référencée par sa position dans le tableau. Exemple :  $[1, 2, \text{test@exemple.ch}, [3, 4, \text{adresse}]]$ .
- Les **tableaux associatifs**. Les tableaux associatifs sont délimités par des accolades rondes {...}. Les valeurs sont séparées par des virgules et référencées par des clés. Exemple :  $\{n: 1, m: 2, c: \text{test@exemple.ch}, \text{tableau: } [3, 4, \text{adresse}]\}$ .

### 2.3.4 Inversion des Valeurs

Par extension des circuits logiques standards, il est possible d'inverser les valeurs reçues en entrées des portes ainsi que celles envoyées aux sorties

Trois types d'inversions sont possibles :

- L'inversion binaire :  $[\neq 0] \rightarrow 0, 0 \rightarrow 1$
- L'inversion additive :  $x \rightarrow -x$
- L'inversion multiplicative :  $x \rightarrow 1/x$

Valeur	Inverse binaire	Inverse additif	Inverse multiplicatif
0	1	0	$\infty$
1	0	-1	1
2	0	-2	0.5
chaîne	0	NaN	NaN
$[2, \{a:0, b:chaîne\}]$	$[0, \{a: 1, b: 0\}]$	$[-2, \{a: 0, b: NaN\}]$	$[0.5, \{a: \infty, b: NaN\}]$

Table 8 Inversions (binaires, additives, et multiplicatives) avec différentes valeurs.

## 2.4 ADRESSES

Les portes actionneurs « écriture » et « lecture » permettent de lire et écrire sur des adresses. Le module logique permet de traiter différents types d'adresses :

- Adresses KNX (Sections 2.4.1 et 2.4.2) : permet de lire, écrire, et écouter des télégrammes KNX sur le bus.
- Adresses GPIO (Section 2.4.3) : permet d'utiliser les entrées et sorties digitales du hardware.
- Adresses Virtuelles (Section 2.4.4) : permet de logger ou stocker des valeurs travaillées en interne.

### 2.4.1 Adresses de Groupe KNX

KNX, aussi appelé Konnex, est un bus de terrain et un protocole d'automatismes pour le bâtiment. Le protocole KNX est un protocole à logique répartie. Contrairement à d'autres protocoles d'automatismes, celui-ci ne fonctionne pas en mode maître/esclave, chaque automate étant indépendant des autres.

Le module logique accède au bus KNX via une passerelle KNX IP ou alternativement via une passerelle KNX USB, les deux types de connexion étant supportés (Figure 10).

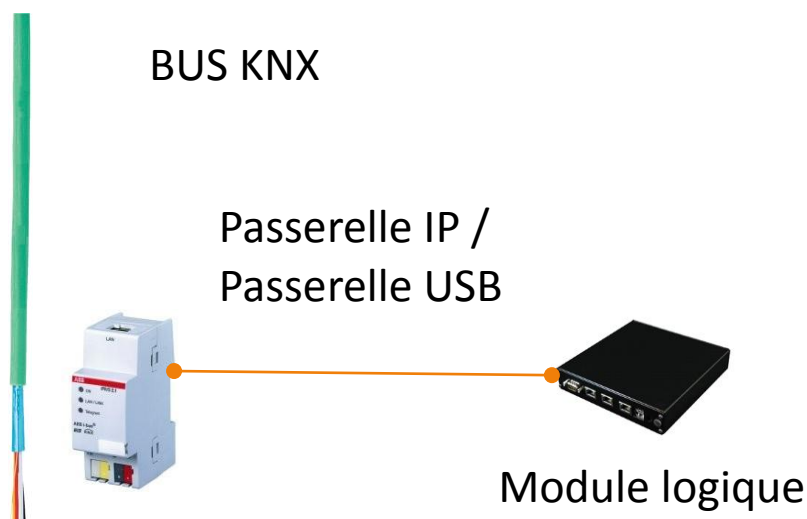


Figure 10 Connexion du module logique au bus KNX via une passerelle KNX IP ou une passerelle KNX USB

La communication entre les appareils est faite avec des télégrammes envoyés sur des adresses de groupe. Celles-ci sont utilisées par les capteurs et les actionneurs afin de communiquer des données ou transmettre des actions. Les adresses de groupe sont codées sur deux bytes. Le module logique utilise la notation structurée sur 3 niveaux main/middle/sub : 5/3/8 bits respectivement (Figure 11).

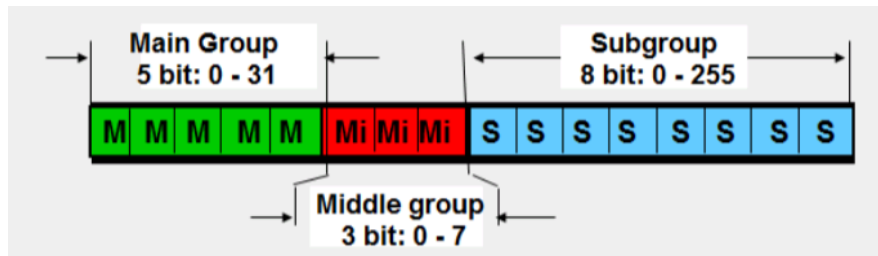


Figure 11 Codage des adresses de groupe KNX à 3 niveaux

Toutes les communications transmises sur le bus entre les différents participants sont interceptées et transmises au module logique qui peut à son tour traiter ses informations, les enregistrer, ou y répondre en écrivant sur le bus. Les opérations suivantes sont supportées sur les adresse de groupe KNX.

- Lecture des télégrammes passant sur le bus.
- Ecriture de télégrammes sur le bus.
- Envoi d'une requête de lecture sur le bus et réception de la réponse.

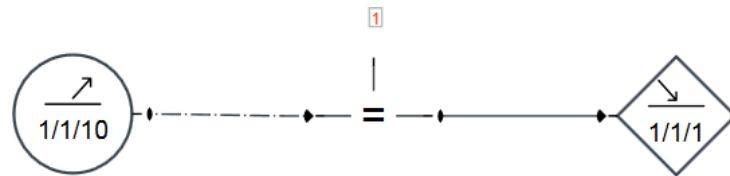


Figure 12 Allume les lumières (adresse de groupe 1/1/1) lorsque la valeur du détecteur de présence (communiquée sur l'adresse de groupe 1/1/10) passe de 0 à 1 (flux montant)

#### 2.4.2 Adresses Physiques KNX

Les capteurs et actionneurs participant sur le bus KNX ont toujours une adresse physique. Une adresse physique est de la forme n.n.n (par exemple 1.1.1) où le nombre représente une topologie réelle. Une adresse physique est allouée à chaque appareil durant la mise en service. La réinitialisation (reset) d'un participant peut se faire depuis un circuit à l'aide de son adresse physique (Figure 13).

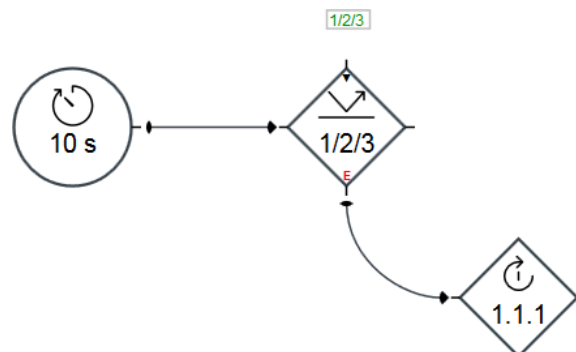


Figure 13 Reset de 1.1.1. Envoie toutes les 10 secondes une requête de lecture concernant l'adresse de groupe 1/2/3 sur le bus KNX. Si aucune réponse n'est reçue, alors la sortie d'erreur (Section 0) est exécutée et l'appareil correspondant (1.1.1) réinitialisé.

### 2.4.3 Adresses GPIO

Si le hardware du module logique dispose de ports **GPIO** (General Purpose Input/Output), comme c'est le cas sur la [raspberry](#), ceux-ci peuvent être utilisés pour communiquer directement sur un circuit électronique. Ces entrées et sorties sont accessibles depuis le module logique via les adresses suivantes:

- **gpi** registre des entrées digitales (8 bits)
- **gpo** registre des sorties digitales (8 bits)

Il est possible de lire l'état des entrées (**gpi**), et concernant les sorties (**gpo**) il est possible d'écrire le registre mais également de lire son état. L'écriture et la lecture des entrées et sorties se fait par des entiers positifs compris entre 0 et  $2^n-1$  où  $n$  représente le nombre de bits à coder (8 pour le registre dans son entier). On peut aussi écrire/lire sur une partie seulement du registre. Par exemple, pour accéder séparément à chacun des bits du **gpo** (numérotés de 0 à 7), les « sous-adresses » suivantes sont à disposition: **gpo(0)** **gpo(1)** **gpo(2)** ... **gpo(7)**. La Figure 14 montre l'usage du GPIO dans un circuit et la Table 9 donne des exemples variés de « sous-adresses » valides pour le **gpi** mettant en évidence les bits concernés.



Figure 14 Si le premier bit du **gpi** (entrées digitales) passe de 0 → 1, alors écris 1 sur tous les bits du **gpo** (sorties digitales). Notez que l'adresse **gpo** est équivalente à la « sous-adresse » **gpo(0:7)**.

Adresse	Contenu du <b>gpi</b>	Valeur binaire	Valeur décimale
<b>gpi(1)</b>	0 0 0 0 0 0 1 0	1	1
<b>gpi(5)</b>	1 1 0 1 1 1 1 1	0	0
<b>gpi(0:7)</b>	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0
<b>gpi(0:7)</b>	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	255
<b>gpi(0:3)</b>	1 1 1 1 0 0 1 0	0 0 1 0	2
<b>gpi(3:0)</b>	1 1 1 1 0 0 1 0	0 1 0 0	4
<b>gpi(0,1,6,7)</b>	1 1 0 0 0 0 0 1	1 1 0 1	13
<b>gpi(2:0,5,7)</b>	1 0 0 0 0 1 0 0	1 0 0 0 1	17

Table 9 « Sous-adresses » valides dérivées du **gpi** (entrées digitales) accompagnées d'un exemple de valeur correspondante.

### 2.4.4 Adresses Virtuelles

Contrairement aux autres adresses, les adresses virtuelles ne sont pas communiquées à l'externe de la machine sur un bus de terrain. Elles sont utiles en interne pour stocker des valeurs, communiquer des valeurs entre différents circuits logiques, ou encore sauvegarder des valeurs dans un historique (fichier de log). Les adresses virtuelles n'ont pas de format particulier : chacune est identifiée par une chaîne de caractères quelconque (voir alias ci-dessous).

### 2.4.5 Alias

Quel que soit la nature d'une adresse (KNX, GPIO, ...), on peut lui définir un alias permettant d'y accéder directement depuis les circuits. Ajouter un niveau d'indirection à l'adressage en recourant aux alias présente des avantages multiples :

- **Clarté** : permet d'explicitier la signification d'une adresse sur le circuit logique. Par exemple si l'adresse de groupe KNX **1/1/1** est paramétrée pour allumer ou éteindre la lumière dans une salle, il est plus parlant d'y faire appel par un alias tel que **lumiere\_salle\_1**.
- **Portabilité** : rend les circuits logiques portables d'un site à un autre, même si les adresses KNX utilisées y sont différentes. Par exemple si la vitesse du vent mesurée par un anémomètre est transmise sur **0/2/0** au site **A** et sur **0/2/3** au site **B**, alors pour des raisons de portabilité il est pratique dans les deux cas d'y accéder par un alias commun « **vent** » (voir Figure 15).
- **Multi-passerelles** : permet de régler des conflits d'adresses lorsque le même module logique fait interagir différentes installations KNX. Par exemple, si le module a accès à deux passerelles KNX différentes : une sur la passerelle **A** et l'autre sur la passerelle **B**, et que sur chacun des deux bus la vitesse du vent mesurée est communiquée sur son adresse de groupe **0/2/0**, alors il est possible de distinguer ces mêmes adresses en leur donnant à chacune un alias différent (voir Figure 16)

Site A		
Alias	Adresse	Description
<b>vent</b>	<b>0/2/0</b>	Vitesse du vent mesurée (m/s)

Site B		
Alias	Adresse	Description
<b>vent</b>	<b>0/2/3</b>	Vitesse du vent mesurée (m/s)

Figure 15 Portabilité. La porte de gauche fait référence à l'adresse de groupe KNX du vent (**0/2/0**). Celle-ci ne fonctionne que sur le site **A**, alors qu'en utilisant un alias comme sur la porte de droite (**vent**), elle peut être utilisée sur les deux sites sans avoir à modifier le circuit.



Passerelle	Alias	Adresse	Description
KNX A	A_vent	0/2/0	Vitesse du vent (m/s)
KNX B	B_vent	0/2/0	Vitesse du vent (m/s)

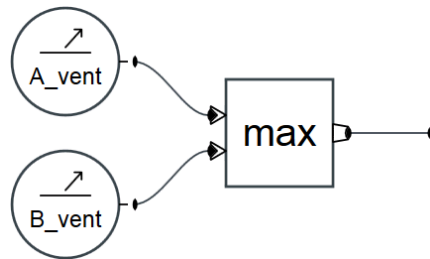


Figure 16 Multi-passerelles. Récupère grâce aux alias *A\_vent* et *B\_vent* des vitesses de vent communiquées avec la même adresse KNX (0/2/0) sur deux bus distincts (passerelles *KNX A* et *KNX B*). Ensuite sélectionne la valeur maximale.

## 2.5 FONCTIONNALITÉS AVANCÉES

Les cas d'utilisation avancés des portes sont abordés dans cette section. Ils nécessitent un peu plus de connaissances et d'efforts, mais leur maîtrise permet de répondre à toutes les situations. Les sujets suivants sont discutés :

- Le mode avancé (Section 2.5.1), qui ajoute sur certaines portes des entrées/sorties et permet ainsi un contrôle plus fin et dynamique de celles-ci.
- La planification de tâches (Section 2.5.2), qui explique comment planifier dans le temps des tâches uniques ou périodiques.
- Les expressions avancées (Section 2.5.2). Cette section montre des façons plus poussées d'utiliser la porte « expression ».
- La porte programmable (Section 2.5.4), qui rend possible la création de ses propres portes en JavaScript.

### 2.5.1 Mode Avancé

Le mode avancé, quand il est activé sur une porte, ajoute des entrées/sorties à celle-ci. Cela permet par exemple de contrôler dynamiquement la porte : l'activer / la désactiver / la mettre en pause / ou encore changer ses paramètres en cours d'exécution. Les portes suivantes ont un mode avancé :

- Les déclencheurs (voir Figure 17)
- Les actionneurs (voir Figure 18, Figure 19)
- La porte délai (voir Figure 20)

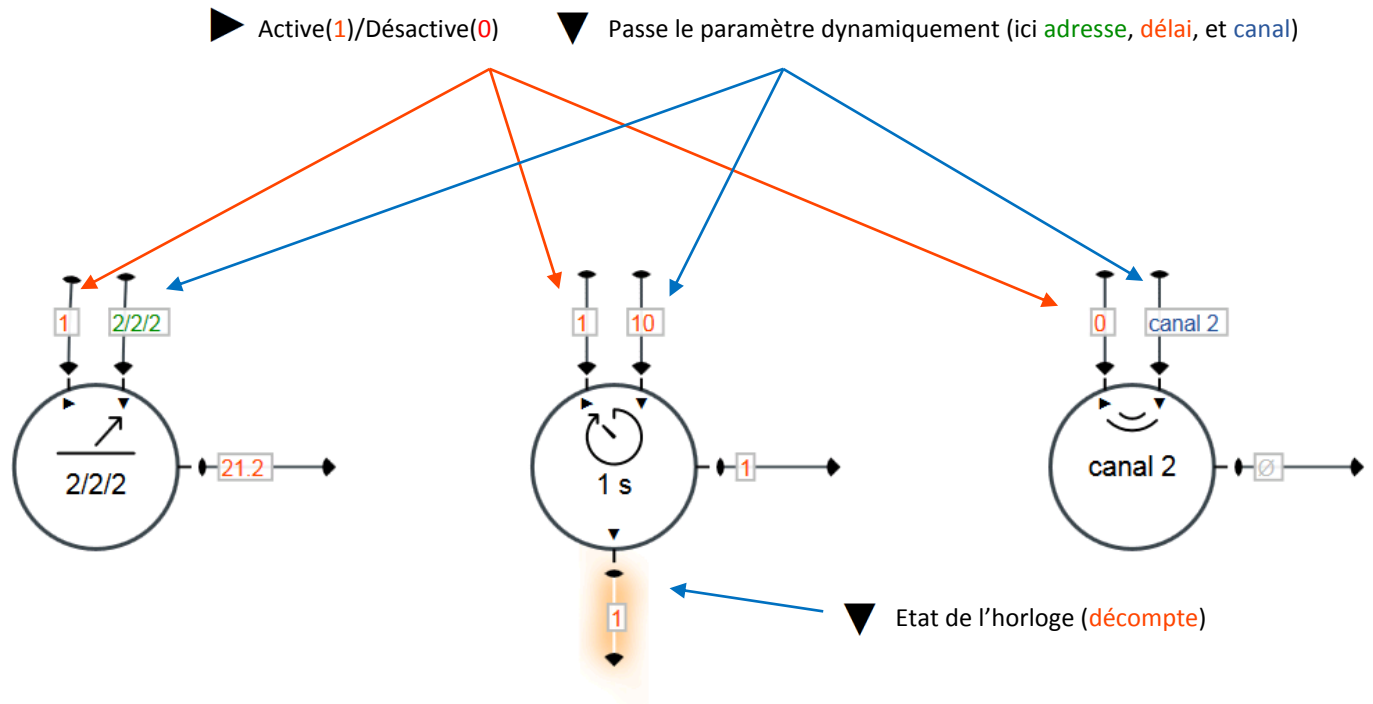


Figure 17 Déclencheurs en mode avancé.

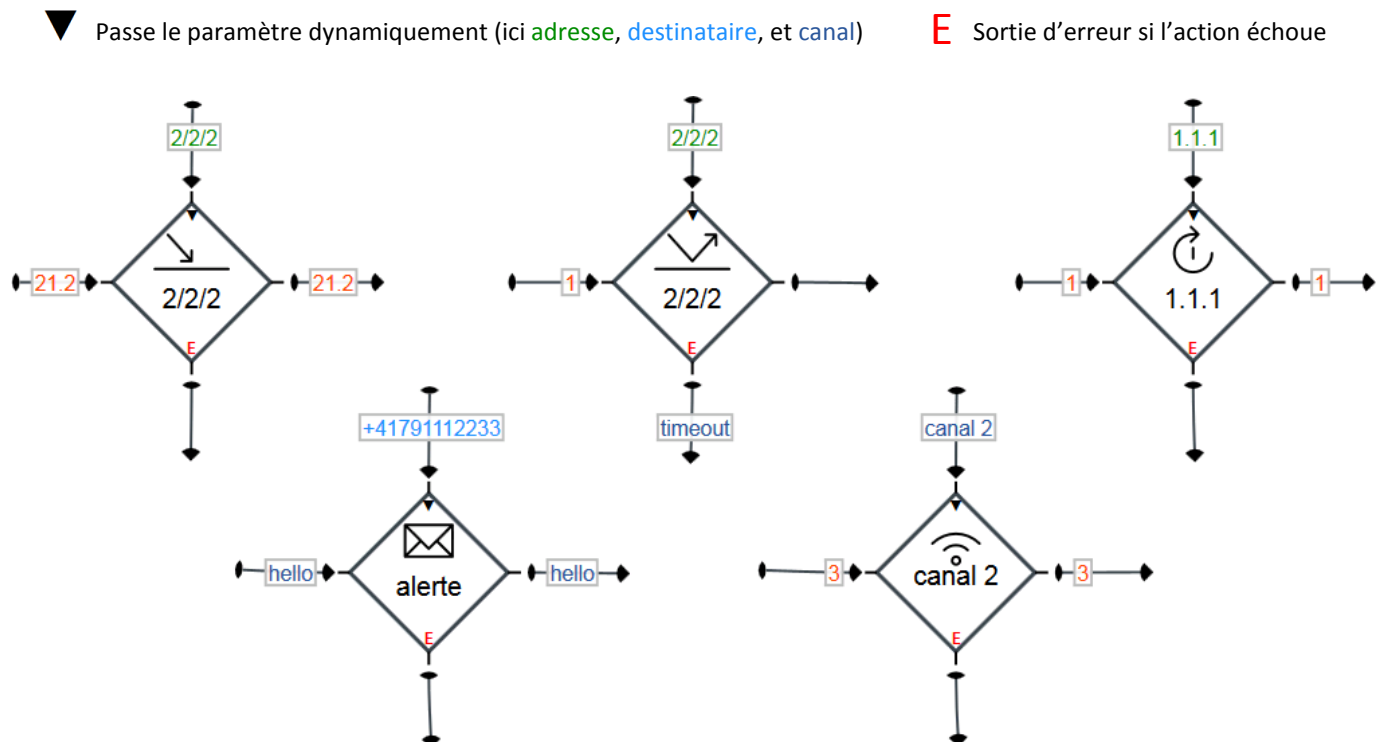


Figure 18 Actionneurs en mode avancé.

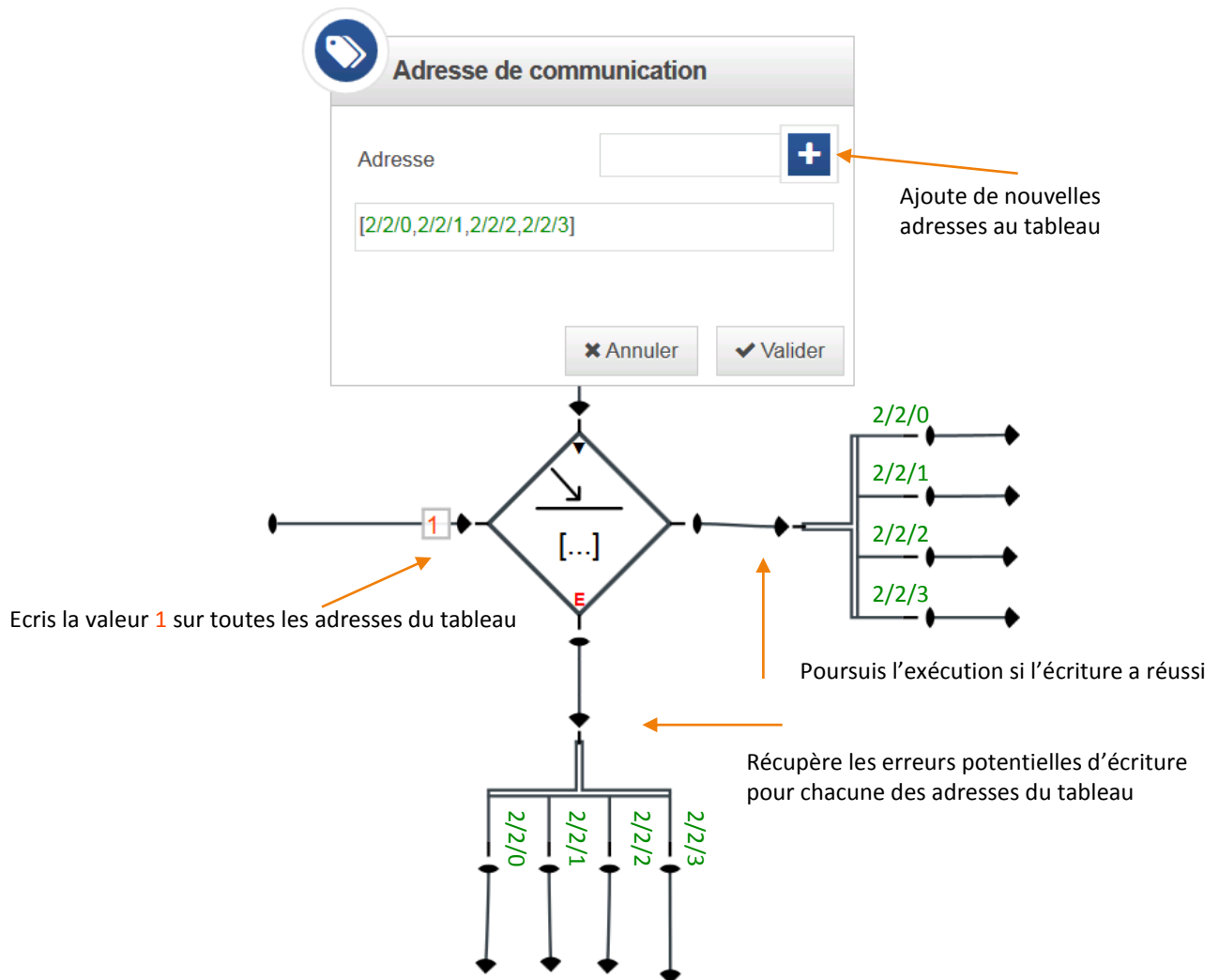


Figure 19 Ecritures multiples. Ecris une valeur (ici 1) sur un nombre multiple d'adresses en utilisant un tableau d'adresses (ici [2/2/0,2/2/1,2/2/2,2/2/3]). A l'aide de la porte « scission » et du mode avancé, on peut récupérer le résultat de l'écriture (succès ou erreur) pour chacune des adresses du tableau.

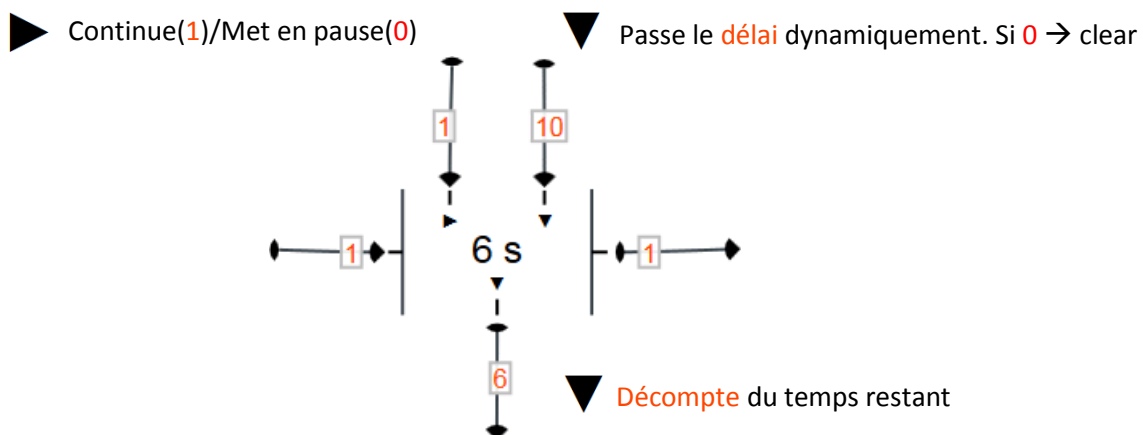


Figure 20 Délai en mode avancé. Permet de mettre en pause et modifier le délai dynamiquement.

## 2.5.2 Planification de Tâches

La planification de tâches se fait à l'aide de la porte « horloge ». La porte horloge comprend différentes sortes différentes de valeurs (voir Figure 21) :

- Les **nombres** (entiers positifs), permettant de répéter une action toutes les **x** millisecondes/secondes/minutes/heures/jours.
- Les **dates** (Table 10), permettant la planification d'une tâche à une date précise.
- Le format **cron** (Table 11, Figure 22), qui est le système Unix de planification de tâches périodiques.

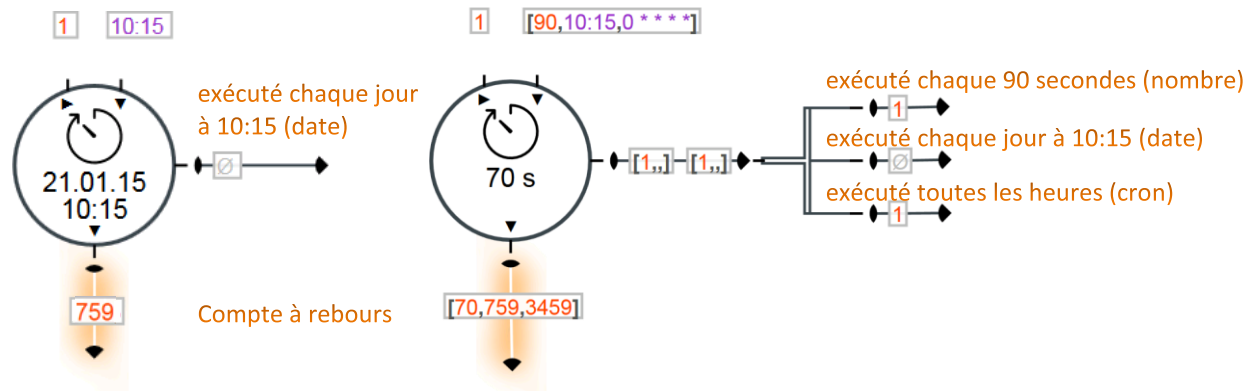


Figure 21 Agende l'exécution de plusieurs tâches. Le premier exemple utilise une valeur simple (10:15). Dans le second exemple un tableau de valeurs ([90,10:15,0 \*\*\*\*]) est utilisé.

Description	Date complète	Date concise
Noel 2016 à midi pile	12:00:00 25.12.2016	12:00 25.12.2016
Nouvel an 2016	00:00:00 01.01.2016	1.1.2016
Chaque nouvel an	00:00:00 01.01.****	1.1.*
Chaque premier du mois	00:00:00 01.**.*****	1.*.*
Chaque jour à 13:00	13:00:00 **.**.*****	13:00
Toutes les heures	**.00:00 **.**.*****	*:0
Toutes les minutes	**.**.00 **.**.*****	*.*
Toutes les 10 minutes	**.0:00 **.**.*****	*.*0
Toutes les secondes	**.**.**. **.**.*****	*.*.*

Table 10 Planification de tâches à l'aide du format de date standard « hh:mm:ss jj.mm.aaaa ». Les tâches répétitives sont planifiées avec le caractère joker « \* ».

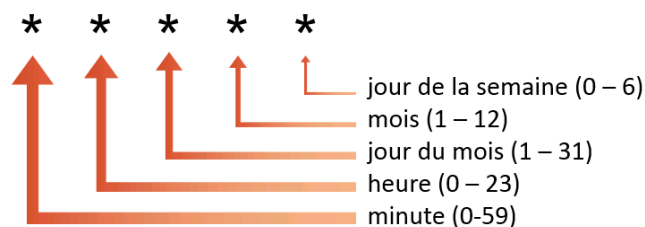


Figure 22 Format cron. Les jours de la semaine sont numérotés de dimanche (0) à samedi (6).

Description	Cron
Exécuté une fois par an à minuit le matin du 1er Janvier	0 0 1 1 *
Exécuté une fois par mois à minuit le matin du premier jour du mois	0 0 1 * *
Exécuté une fois par semaine à minuit le dimanche matin	0 0 * * 0
Exécuté une fois par jour à minuit	0 0 * * *
Exécuté une fois par heure	0 * * * *
Exécuté une fois par minute	* * * * *
Exécuté toutes les 5 minutes	*/5 * * * *

Table 11 Exemples de cron.

### 2.5.3 Expressions Avancées

La porte expression ne se limite pas aux simples expressions calculatoires basiques tel  $(a+b)/2$ . Une expression est en réalité une ligne de code évaluée par le moteur JavaScript. Ceci rend possible beaucoup d'opérations sur les valeurs :

- Opérations logiques et comparaisons: Par exemple  $(a > b \ \&\& \ a > c) \ || \ a == 0$  retourne **1** si  $a$  est plus grand que  $b$  et  $c$ , ou que  $a$  est égal à  $0$ . Autrement retourne **0**.
- Opérateur ternaire :  $a == b ? 10 : -10$ . Retourne **10** si  $a$  est égal à  $b$ . Sinon retourne **-10**
- Appel aux [fonctions mathématiques JavaScript natives](#) :  $\text{acos}(x)$   $\text{asin}(x)$   $\text{atan}(x)$   $\text{atan2}(y,x)$   $\text{ceil}(x)$   $\text{cos}(x)$   $\text{exp}(x)$   $\text{floor}(x)$   $\text{log}(x)$   $\text{max}(x,y,z,...,n)$   $\text{min}(x,y,z,...,n)$   $\text{pow}(x,y)$   $\text{random}()$   $\text{round}(x)$   $\text{sin}(x)$   $\text{sqr}(x)$  et  $\text{tan}(x)$
- Appel aux [fonctions des chaînes de caractères](#) :  $\text{charAt}()$   $\text{concat}()$   $\text{indexOf}()$   $\text{lastIndexOf}()$   $\text{match}()$   $\text{replace}()$   $\text{search}()$   $\text{slice}()$   $\text{split}()$   $\text{substr}()$   $\text{substring}()$   $\text{toLowerCase}()$   $\text{toUpperCase}()$   $\text{trim}()$  ...
- Utilisation d'[expressions régulières](#) :  $\text{/raclette}s+magistrale/.test("Je mange une raclette magistrale")$  retourne **1**.
- Appel aux fonctions manipulant les [tableaux](#) :  $\text{concat}()$   $\text{every}()$   $\text{filter}()$   $\text{join}()$   $\text{map}()$   $\text{reduce}()$   $\text{reduceRight}()$   $\text{reverse}()$   $\text{slice}()$   $\text{some}()$   $\text{sort}()$  ... La Figure 23 illustre la manipulation d'un tableau de valeurs à l'aide d'expressions.
- ...

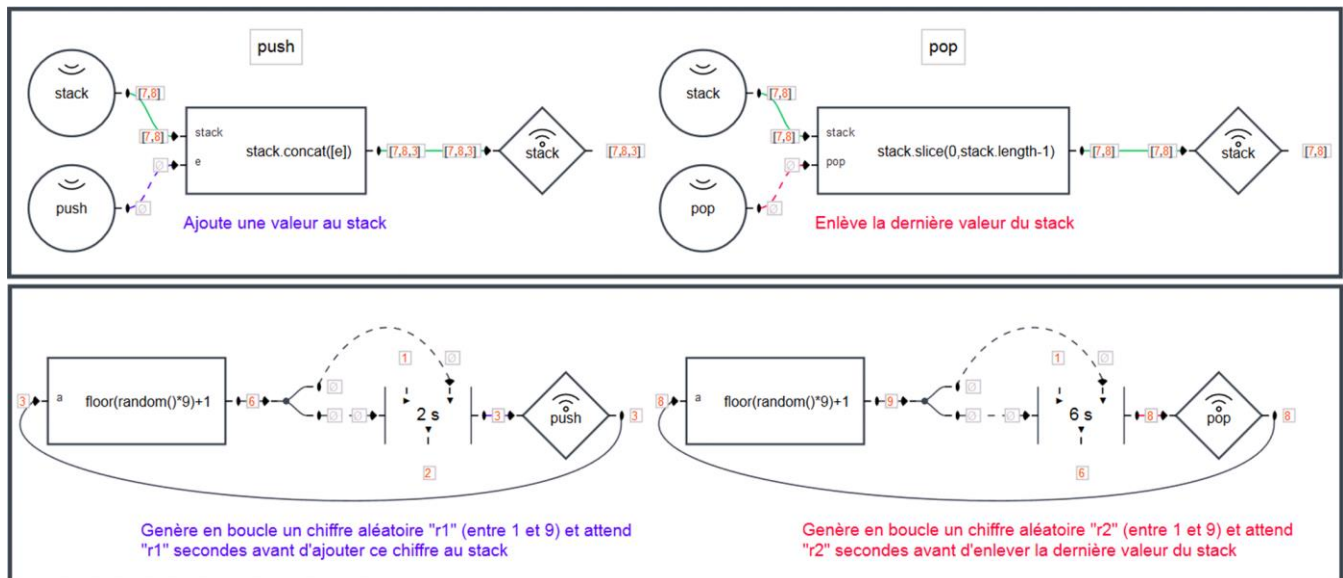


Figure 23 La partie du haut présente une façon intéressante d'implémenter un « stack » : une pile de valeurs avec une fonction « push » empilant une nouvelle valeur, et une fonction « pop » jetant la dernière valeur. Dans la partie du bas des valeurs sont aléatoirement empilées sur le stack et ensuite jetées du stack. Notez que les portes « expressions » se servent des fonctions JavaScript natives `concat()`, `slice()`, `floor()`, et `random()` afin de construire le stack et générer les chiffres aléatoires.

#### 2.5.4 Porte Programmable (JavaScript)

Créer ses propres portes personnalisées via le langage de programmation JavaScript offre beaucoup de puissance et de flexibilité. De plus les portes développées sont ensuite réutilisables dans différents projets via les fonctionnalités d'import/export de l'interface.

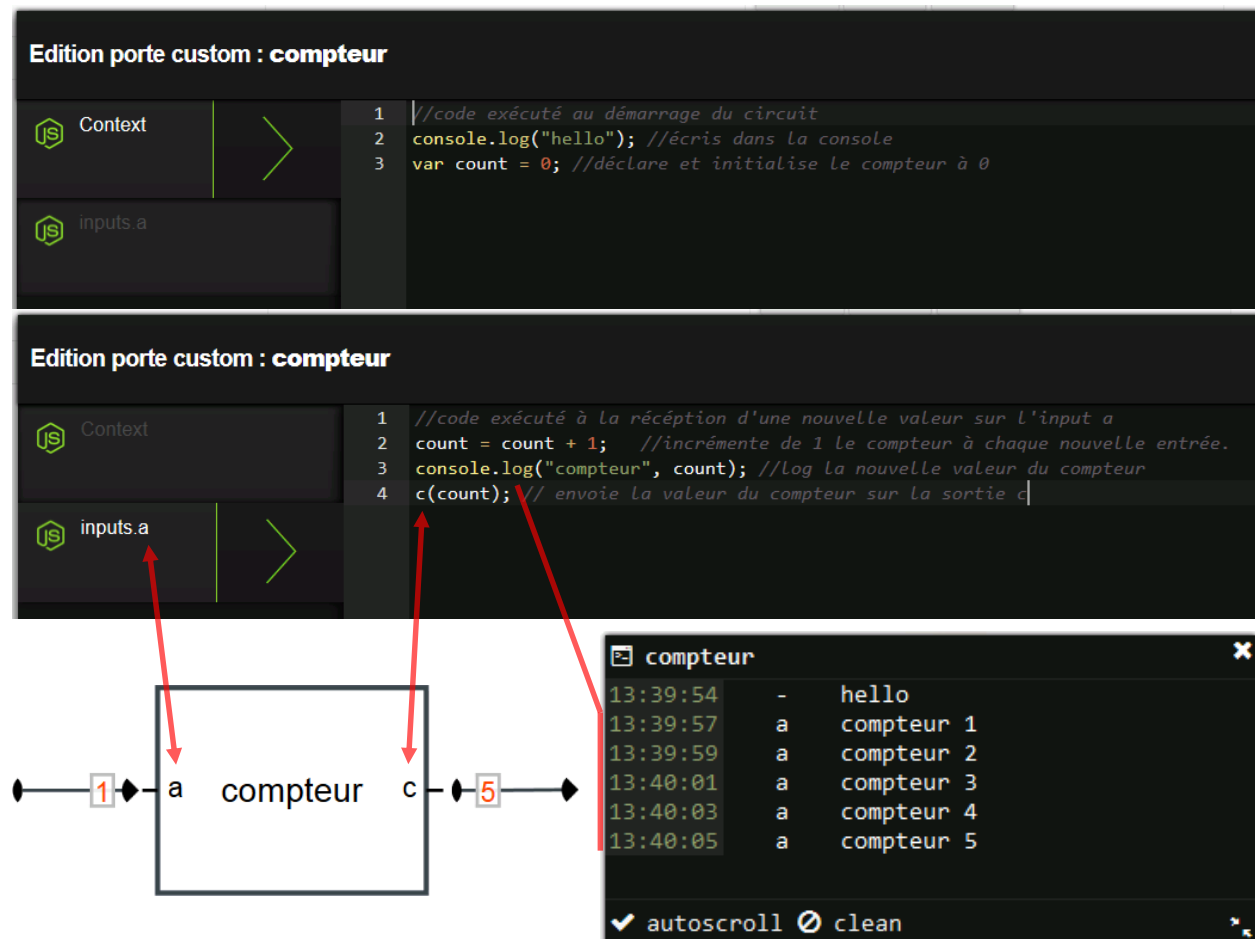
- Nombre illimité d'entrées et sorties.
- Nombre illimité de paramètres éditables.
- Accès aux fonctionnalités [JavaScript](#) natives (Object, Array, Buffer, Math, string, ...).
- Accès aux modules [node.js](#) suivants : [assert](#), [buffer](#), [crypto](#), [dgram](#), [dns](#), [domain](#), [events](#), [http](#), [https](#), [later](#), [net](#), [os](#), [path](#), [stream](#), [string\\_decoder](#), [suncalc](#), [tty](#), [url](#), [util](#), [zlib](#).
- Editeur de code [Ace](#) intégré avec détection automatique d'erreurs et auto-complétion.
- Console pouvant afficher des infos utiles au débogage.

La Figure 24 montre l'implémentation basique d'un compteur, et la Table 12 explique comment accéder à la valeur d'une entrée et exécuter une sortie depuis le code de la porte programmable.

La Figure 25 (régulateur PID) et la Figure 26 (Pulse Width Modulation) présentent deux exemples d'utilisation de portes programmables dans le cadre de la régulation de température d'une pièce.

La Figure 27 illustre une porte utile en domotique qui calcule le moment du lever et coucher du soleil à partir de la latitude et longitude. Cette porte est pratique pour exécuter des actions aux moments particuliers de la journée (lever de stores le matin, baisser les stores le soir, allumer des lumières, fermer un portail, etc...). L'implémentation (Table 13) utilise le module [suncalc](#) pour effectuer les calculs solaires. Note: les calculs ne tiennent pas en compte le relief pouvant par exemple retarder le lever du

soleil de quelques minutes. Si c'est le cas un décalage peut être ajouté grâce à une porte « délai » (Figure 28) ou directement dans le code.



**Edition porte custom : compteur**

Context

```

1 //code exécuté au démarrage du circuit
2 console.log("hello"); //écris dans la console
3 var count = 0; //déclare et initialise le compteur à 0

```

inputs.a

**Edition porte custom : compteur**

Context

```

1 //code exécuté à la réception d'une nouvelle valeur sur l'input a
2 count = count + 1; //incrémente de 1 le compteur à chaque nouvelle entrée.
3 console.log("compteur", count); //log la nouvelle valeur du compteur
4 c(count); // envoie la valeur du compteur sur la sortie c

```

inputs.a

compteur

```

13:39:54 - hello
13:39:57 a compteur 1
13:39:59 a compteur 2
13:40:01 a compteur 3
13:40:03 a compteur 4
13:40:05 a compteur 5

```

autoscroll clean

Figure 24 Implémentation d'un compteur. En double cliquant sur la porte depuis l'interface, l'éditeur de code s'ouvre. Dans l'onglet « Context » se trouve du code évalué à l'initialisation du circuit. Les autres onglets contiennent du code invoqué à la réception d'une nouvelle valeur sur l'entrée correspondante (ici « a »). Note : la fonction `console.log()` permet de laisser une trace lors de l'exécution dans une fenêtre de log rattachée à la porte.

Accède à la valeur de l'entrée « a »	Exécute 1 sur la sortie « c »
<code>Gate.inputs.a.value</code>	<code>Gate.outputs.c.exec(1)</code>
<code>inputs.a.value</code>	<code>outputs.c.exec(1)</code>
<code>a</code>	<code>c(1)</code>

Table 12 : Trois façons alternatives d'accéder à la valeur d'une entrée et d'exécuter une sortie depuis le code d'une porte programmable.

Context

inputs.mesure

inputs.consigne

inputs.Kp

inputs.Ki

inputs.Kd

```

1 //code exécuté au démarrage du circuit
2 //initialise les variables nécessaires aux calculs du régulateur PID
3 var dt = 1; //delta de temps entre chaque correction (en secondes)
4 var integrale = 0;
5 var erreurPrecedente = 0;
6
7 setInterval(function(){ //répète la régulation toutes les 'dt' secondes
8   //calcul de l'erreur
9   var erreur = consigne - mesure;
10
11   if(!isNaN(erreur)){ // si l'erreur est un nombre valide,
12     // alors calcul l'intégrale et la dérivée
13     integrale = integrale + erreur*dt; //facteur intégral (somme des erreurs)
14     var derivee = (erreur - erreurPrecedente)/dt; //facteur dérivatif
15     erreur_precedente = erreur;
16
17     //Calcul la correction (P + I + D) à effectuer sur l'actionneur
18     var PID = Kp*erreur + Ki*integrale + Kd*derivee;
19
20     if(!isNaN(PID)){ //si la correction est un nombre valide,
21       // alors on l'envoie sur la sortie pour l'actionneur
22       correction(PID);
23     }
24   }
25 },dt*1000);
26

```

Réglage du PID

0.1 0 0

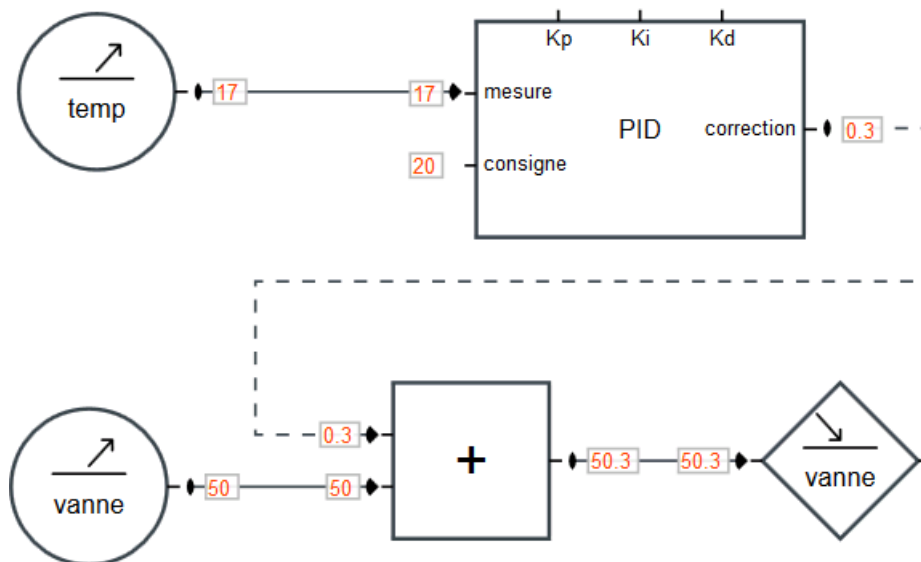


Figure 25: Implémentation simple d'un régulateur PID (actions Proportionnelle, Intégrale, et Dérivée). Dans cette boucle de régulation, le **PID** ajuste l'ouverture de la vanne (+0.3%) afin que la température mesurée dans la pièce (17°C) approche la température de consigne (20°C).



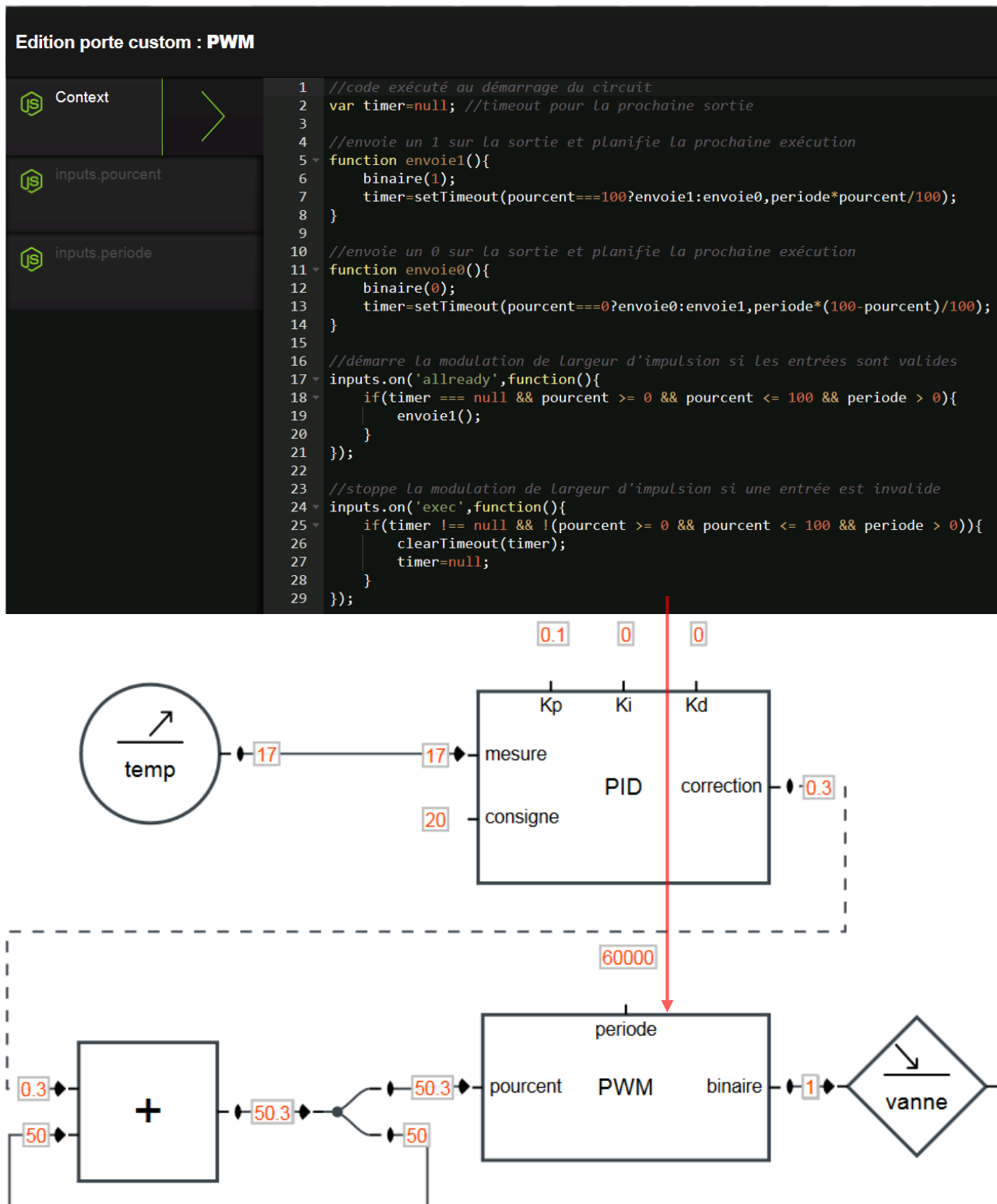


Figure 26: Implémentation d'une porte **PWM (Pulse Width Modulation)**. Dans cette boucle de régulation, la porte **PWM** permet d'ouvrir (1) / fermer (0) la vanne de chauffe durant un certain laps de temps défini par la période (ici 60 secondes) et le pourcentage (ici 50.3%) du temps où la vanne doit rester ouverte. La modulation de largeur d'impulsion est pratique dans le cas où le degré d'ouverture de la vanne n'est pas réglable et n'accepte qu'une valeur binaire : ouvert ou fermé.

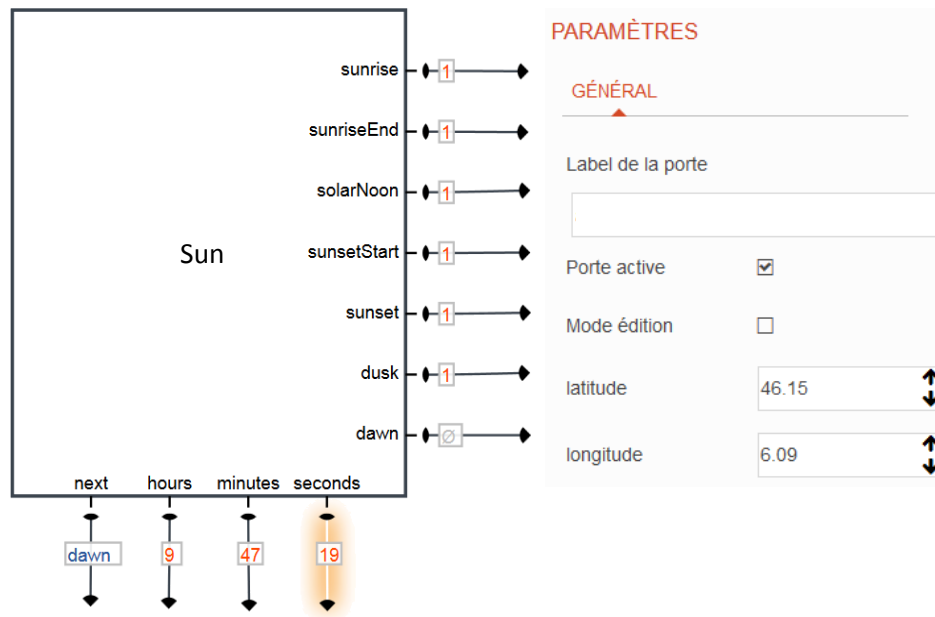


Figure 27 Porte « Sun ». Des exécutions (sorties de droite) sont déclenchées lorsque le soleil se lève (sunrise), est au plus haut (solarNoon), se couche (sunset), etc... les sorties du bas indique le prochaine évènement et le temps restant (ici l'aube est prévue dans 9 heures 47 minutes et 10 secondes). Afin de pouvoir effectuer les calculs deux paramètres ont été ajoutés à cette porte : la latitude et la longitude (ici 46°12'N, 06°09'E correspondant à la localisation de Genève).

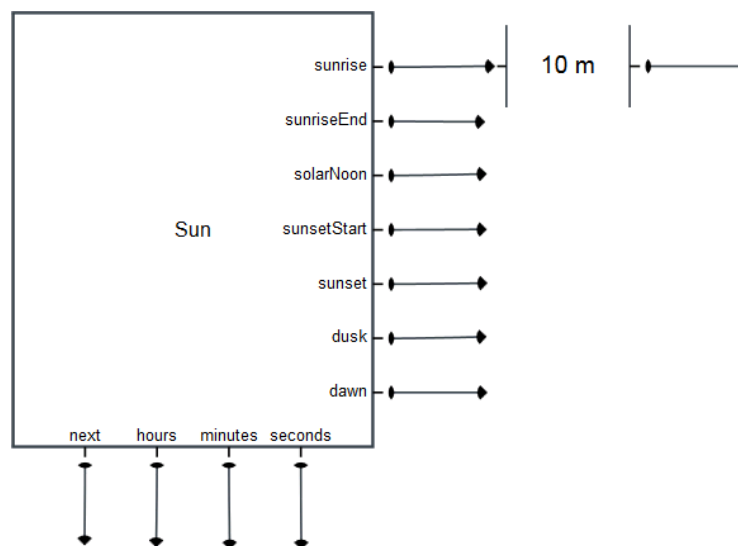


Figure 28 Porte « Sun ». Ici en raison du relief un délai de 10 minutes est ajouté au calcul du lever de soleil.

```

var suncalc = require('suncalc'); //https://github.com/mourner/suncalc
var nextTimer = null;

//schedules execution of next event (sunrise/sunset/...)
function scheduleNextEvent(startTs) {
    var now = Date.now();
    //compute events timestamps based on latitude and longitude parameters
    var res = suncalc.getTimes(startTs ? startTs : now, latitude, longitude);
    var keys = outputs.filter(function(o) {return o.position ==
'right'}).map(function(o) {return o.name});
    var tss = keys.map(function(k) {return res[k].getTime()});
    var tsMin = tss.map(function(ts) {return ts < now ? Number.MAX_VALUE :
ts}).reduce(function(a,b) {return Math.min(a,b)});

    if(tsMin !== Number.MAX_VALUE) { //timestamp of the next event found
        var indexMin = tss.indexOf(tsMin);
        var output = outputs[keys[indexMin]];

        nextTimer = setTimeout(function() { //executed when the event happens
            output.exec(1); //send an impulsion to the correct output.
            scheduleNextEvent();
        }, tsMin-now);

        next(keys[indexMin]); //indicates the next event (sunrise/sunset/...)
    }
    else { //all the events (sunrise/sunset/..) are past for today
        scheduleNextEvent(now+1000*60*60*24); //find tomorrow next event
    }
}

//countdown until next event (sunrise/sunset/...)
function countdown() {
    var tsl = Math.round(nextTimer.next()/1000); // total seconds left
    var s = tsl % 60; // seconds left
    var m = Math.floor(tsl/60) % 60; //minutes left
    var h = Math.floor(tsl/3600); // hours left
    seconds(s); //outputs the seconds left
    if(this.m !== m) { //outputs the minutes left (when it changes)
        this.m = m;
        minutes(m);
    }
    if(this.h !== h) { //outputs the hours left (when it changes)
        this.h = h;
        hours(h);
    }
}

scheduleNextEvent(); //start at initialization

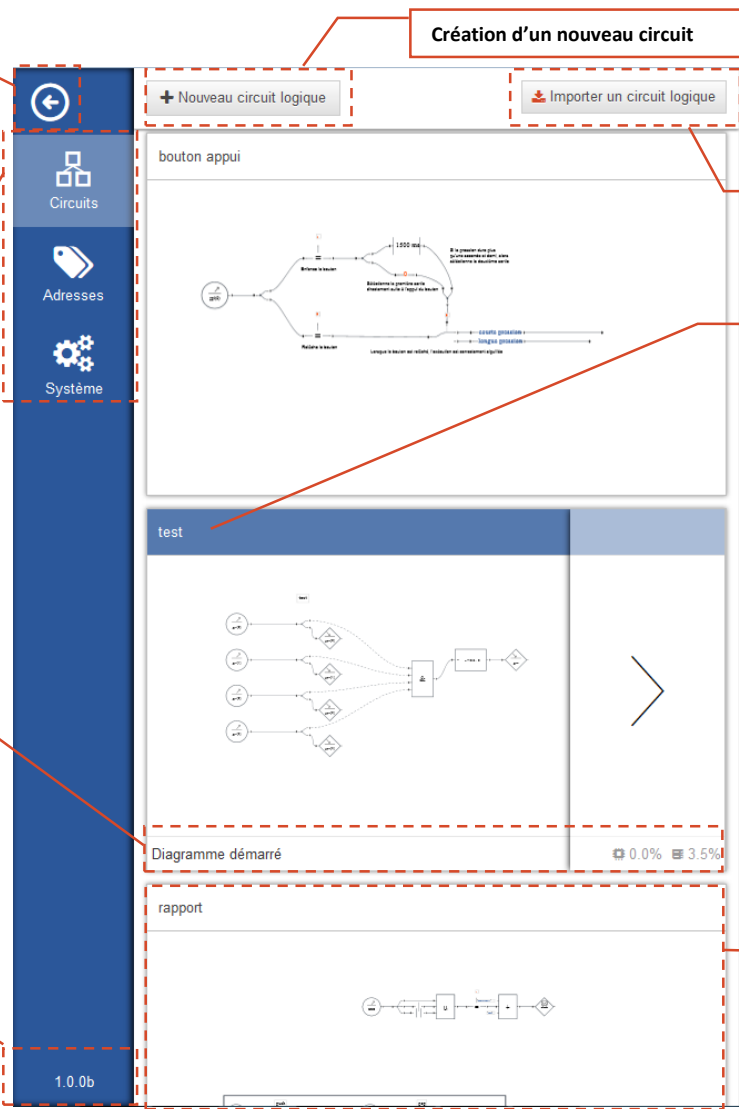
countdown(); //update countdown at initialization
setInterval(countdown, 1000); // update countdown every seconds

```

Table 13 Implémentation de la porte « Sun » (calcul des levers et couchers de soleil) à l'aide du module nodejs suncalc. Ce code est exécuté une fois au démarrage du circuit.

## 3 INTERFACE

### 3.1 PANEL DE SÉLECTION



**Sortir du menu (revient à l'interface d'édition du circuit.)**

**Création d'un nouveau circuit**

**Importer un circuit (fichier .WBL)**

**Menu de navigation:**

- Circuits**  
Edition des circuits logiques
- Adresses**  
Paramétrage des adresses (KNX, GPIO, Virtuelles)
- Système**  
Paramètres généraux, réglage du réseau, réglage de la date et heure, utilisateurs, et mises à jour du système.

**Circuit en cours d'édition**

La miniature du circuit logique avec l'entête bleu signifie que ce circuit est en cours d'édition.

**Statut du circuit logique**

Si le circuit est en fonction, alors il est possible de voir son utilisation du processeur et de la mémoire de ce dernier (en % des ressources totales).

**Version actuelle du système**

Affichage la version actuelle du système. Lorsqu'une mise à jour est disponible, elle sera affichée ici.

**Miniature du circuit**

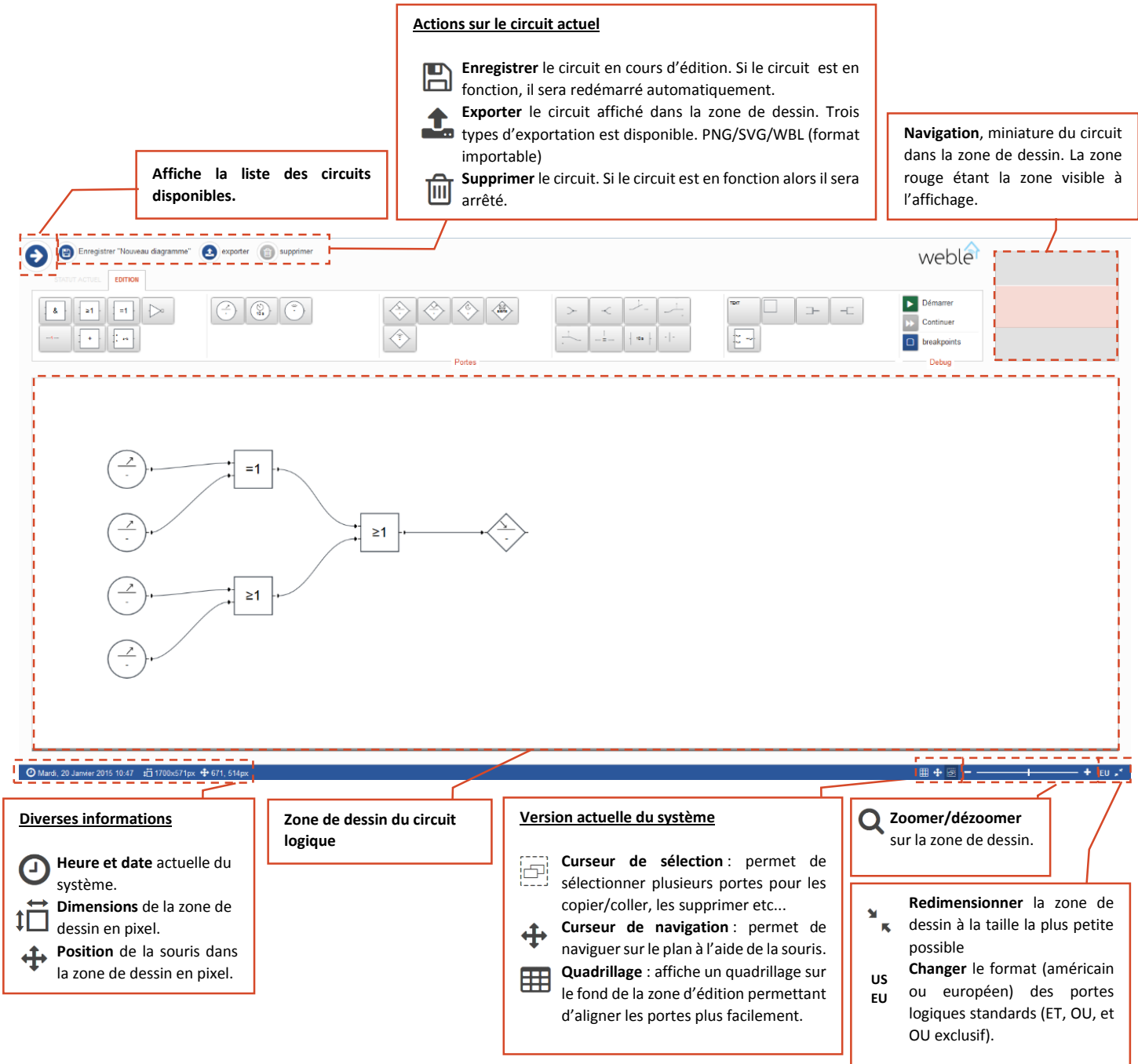
Lorsqu'un circuit est sauvegardé, une miniature de ce dernier est créée.

**Diagramme démarré**

**rapport**

1.0.0b

## 3.2 LAYOUT



**Affiche la liste des circuits disponibles.**

**Actions sur le circuit actuel**

- Enregistrer** le circuit en cours d'édition. Si le circuit est en fonction, il sera redémarré automatiquement.
- Exporter** le circuit affiché dans la zone de dessin. Trois types d'exportation est disponible. PNG/SVG/WBL (format importable)
- Supprimer** le circuit. Si le circuit est en fonction alors il sera arrêté.

**Navigation**, miniature du circuit dans la zone de dessin. La zone rouge étant la zone visible à l'affichage.

**Diverses informations**

- Heure et date** actuelle du système.
- Dimensions** de la zone de dessin en pixel.
- Position** de la souris dans la zone de dessin en pixel.

**Zone de dessin du circuit logique**

**Version actuelle du système**

- Curseur de sélection** : permet de sélectionner plusieurs portes pour les copier/coller, les supprimer etc...
- Curseur de navigation** : permet de naviguer sur le plan à l'aide de la souris.
- Quadrillage** : affiche un quadrillage sur le fond de la zone d'édition permettant d'aligner les portes plus facilement.

**Zoomer/dézoomer** sur la zone de dessin.

**Redimensionner** la zone de dessin à la taille la plus petite possible

**Changer** le format (américain ou européen) des portes logiques standards (ET, OU, et OU exclusif).

### 3.3 EDITION

#### Edition

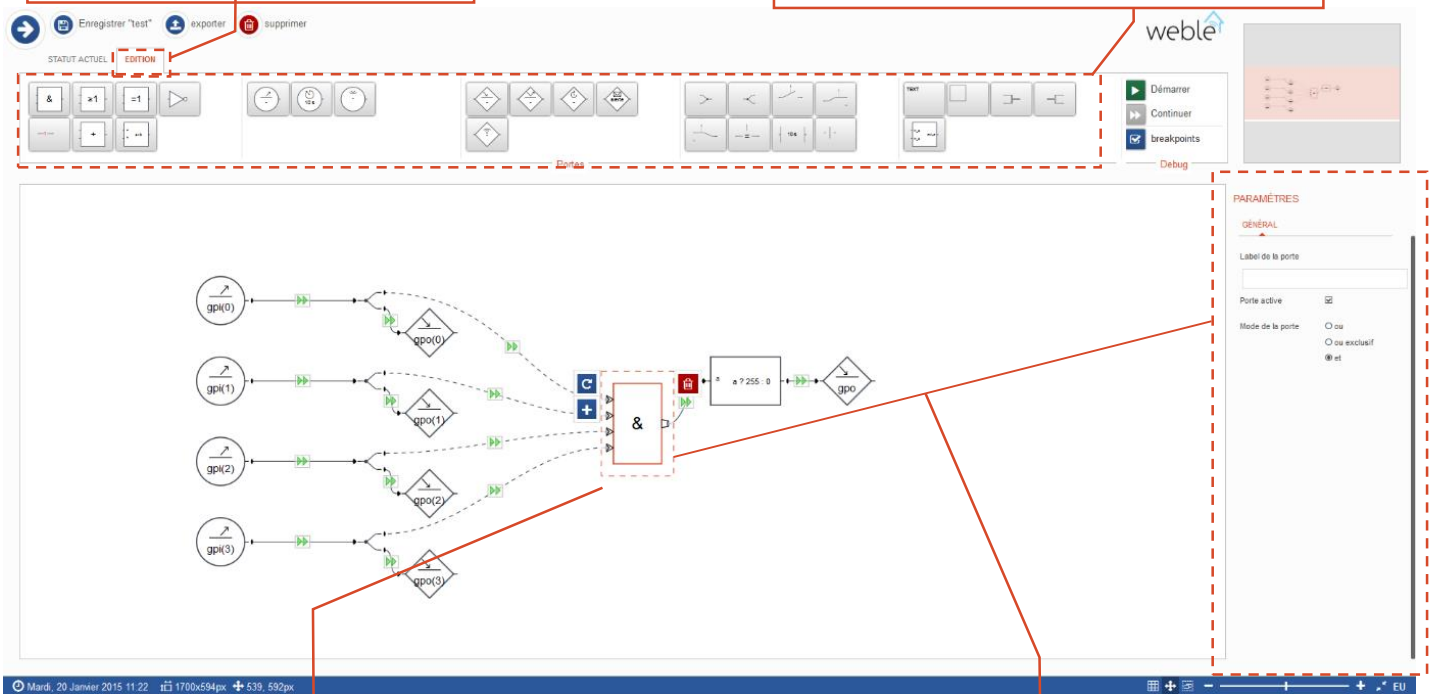
La vue édition permet d'éditer le circuit, d'y ajouter des portes, de les paramétrer etc...

Elle intègre également un débbugger permettant de tester son circuit.

#### Liste des portes

Les portes disponibles sont classées par catégorie : Porte logiques, déclencheurs, actionneurs, contrôleurs de flux et autres.

Pour ajouter une porte, cliquez simplement sur une d'entre elles et placez la sur la zone de dessin puis de relier à l'emplacement souhaité pour la déposer.



#### Sélection des portes

Pour sélectionner une porte, il suffit de cliquer dessus. La porte devient alors rouge et est encadrée. Certaines actions de sélection sont communes à toutes les portes, alors que d'autres actions sont spécifiques à la porte sélectionnée.

**Faire pivoter la porte.** Toutes les portes disposent de cette action. Elles peuvent ainsi être pivotées (de 90° à chaque fois) afin de rendre plus clair le circuit.

**Ajouter des sorties/entrées.** La plupart des portes autorisent l'ajout/suppression d'entrées ou de sorties. Par convention, les sorties se trouvent à droite d'une porte, et les entrées à gauche. Si ce bouton se trouve sur la droite de la porte alors une sortie sera ajoutée au moment où vous cliquerez dessus.

**Supprimer la porte.** Toutes les portes peuvent être supprimées à l'aide de ce bouton. Une fois sélectionnée, vous pouvez aussi la supprimer avec la touche « del/Delete » de votre clavier.

#### Paramétrage des portes

Une fois la porte sélectionnée ses paramètres s'affichent sur le panel de droite. Deux paramètres sont communs à toutes les portes :

**Le label** affiche un texte au-dessus de la porte. Ce texte est utile uniquement à titre informatif.

**Active/désactive** la porte permettant si besoin de la rendre complètement inopérante dans le circuit.

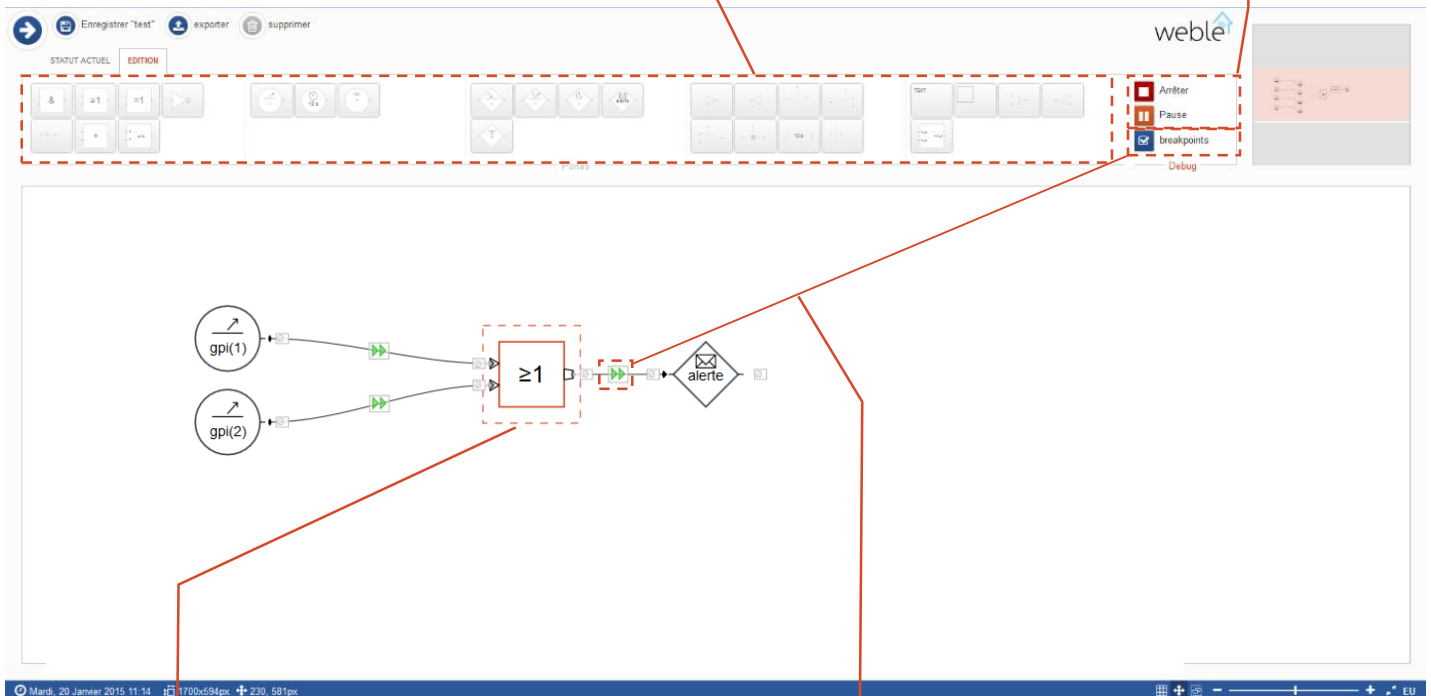
## 3.4 DÉBUG

### Ajout de nouvelles portes impossible

Lorsque le débogueur est démarré, il est impossible d'ajouter de nouvelles portes. Cliquer sur la zone des catégories des portes arrête automatiquement le débogueur.

### Debug

- ▶ **Démarrer/stopper** le débogueur. Ceci permet de tester le diagramme avant de le mettre en production. A l'aide du débogueur il est possible de mettre sur pause le circuit, ce qui permet de visualiser et trouver les erreurs facilement.
- ▶▶ **Continuer/mettre sur pause** : Lorsque le débogueur est démarré, il est possible de mettre sur pause le circuit à n'importe quel moment en cliquant sur ce bouton.



### Portes déplaçables mais non paramétrable

Lorsque le débogueur est activé, il est possible de déplacer les portes sur la zone de dessin mais pas d'éditer leurs paramètres.

### Breakpoints

- ☒ **Afficher/cacher breakpoints** : Permet d'afficher ou cacher les « breakpoints » sur les liens, mettant ainsi le circuit sur pause pendant l'exécution du lien concerné.
- ☐

## 3.5 STATUT

### Statut

La vue « statut » permet de gérer et visualiser le statut du circuit.

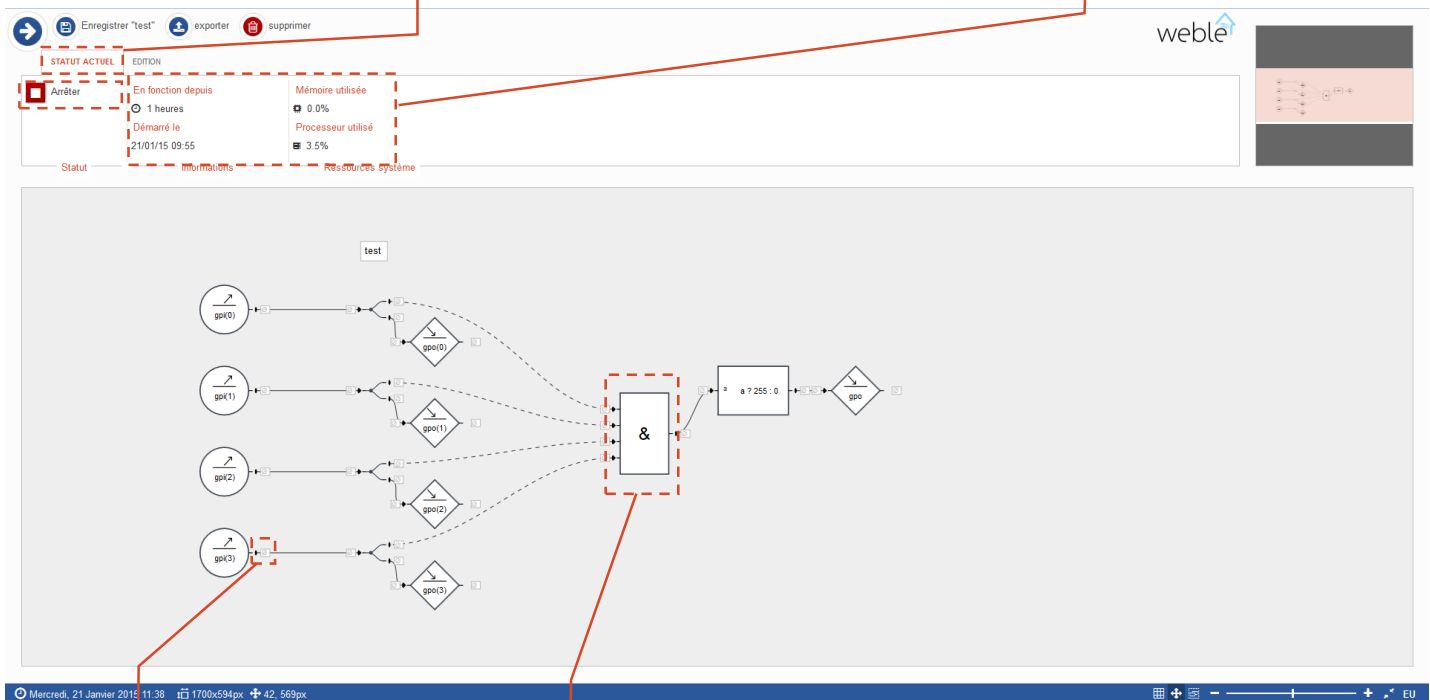
Depuis le menu on peut démarrer (déployer pour la production) le circuit et ensuite si nécessaire l'arrêter.

Une fois le circuit démarré, son état d'exécution actuel est reproduit en temps réel sur le dessin. Ceci permet de s'assurer de son bon fonctionnement à tous moments.

**Note :** si des modifications/corrections sont enregistrées alors que le circuit est présentement en fonction, alors celui-ci sera redémarré automatiquement afin de les prendre en compte.

### Information fonctionnent circuit

Lorsque le circuit est en fonction, les informations relatives à son temps de fonctionnement ainsi qu'aux ressources machines utilisées par ce dernier s'affiche dans cette zone.



### Changement de valeur

Lorsque le circuit est en fonction, il est possible d'entrer une valeur et forcer son exécution sur l'entrée ou la sortie sélectionnée. Ceci permet d'exécuter manuellement certaines portions du circuit.

### Paramétrage des portes impossible

Depuis cette vue il est impossible de déplacer les portes et d'éditer leurs paramètres. Seule la navigation dans la zone de dessin est possible.