

Compteurs, variables et afficheurs dans Automgen

Domaine d'application :
Traitement programmé de l'information

Type de document :
Travaux Pratiques

Classe :
Première

Date :

☞ Mise en situation et objectifs du TP ☞

L'objectif de ce TP est de découvrir l'utilisation des **compteurs** et des **variables utilisateurs** dans Automgen, que ce soit pour la conception d'un système logique combinatoire (sous forme d'un logigramme) ou pour un système logique séquentiel (sous forme d'un grafcet). Les notions abordées dans ce TP sont :

- * Utilisation des **compteurs** dans Automgen
- * Utilisation des **variables utilisateurs**
- * Utilisation d'une **dynavue** pour la mise au point du programme
- * Ajout d'un **afficheur** sur le pupitre pour visualiser la valeur d'une variable numérique

Ce TP est structuré en 2 parties :

- * La **partie apprentissage** vous explique les principes de base (liste non exhaustive) à connaître
- * La **partie application** dans laquelle vous devez proposer une solution à une problématique donnée

☞ Travail demandé ☞

I - Partie apprentissage

Cette première partie va vous montrer des nouvelles syntaxes, des nouveaux principes et des nouveaux opérateurs utilisables dans votre projet Automgen, que votre programme soit sous forme d'un logigramme ou d'un grafcet. Toutes les informations données dans cette première partie ne seront ni rappelées dans la partie application, ni rappelées dans les futurs TP Automgen et doivent être mémorisées afin de les utiliser dans les TP futurs

Il vous appartient donc de retenir le contenu de cette première partie et **de ressortir de vous-même ce document** dans les TP futurs en cas de besoin : ce TP vous servira par la suite de document ressource pour Automgen.

I - 1 - Syntaxe du logiciel Automgen

Automgen est un atelier d'automatisme, de supervision et de simulation interactive des systèmes. Il permet de :

- * Créer un système logique **combinatoire** dont le fonctionnement est décrit par un **logigramme**
- * Créer un système logique **séquentiel** dont le fonctionnement est décrit par un **grafcet**

Automgen permet d'utiliser différents types de variables (des entrées, des sorties, des temporisations, etc.), ayant chacun un rôle bien défini. La syntaxe de chaque type de variable est normalisée selon le tableau suivant :

Type de variable	Syntaxe Automgen
les entrées	i0 à i23
les sorties	o0 à o15
les temporisations	t0 à t9999
les étapes du grafcet	x0 à x9999
les compteurs	c0 à c9999
les variables utilisateur	m200 à m9999

Remarques :

- * Les variables **m0** à **m199** sont réservés pour le système et sont appelés **les variables système**
- * Pour programmer votre application seules les variables à partir de **m200** sont disponibles
- * Les variables **m200** à **m9999** sont appelées **les variables utilisateur**
- * La table des symboles permet de personnaliser le nom de tout type de variable
- * Automgen ne distingue pas les majuscules et les minuscules dans les noms des variables ou des symboles
- * Les noms normalisés (i0, o2, t3, x7, m204, c9, etc.) sont affichés en **VIOLET** sur le folio
- * Les symboles personnalisés (créés dans la table des symboles) sont affichés en **VERT** sur le folio
- * Les opérateurs [+ . / etc.] sont affichés en **BLEU** sur le folio
- * Les commentaires sont affichés en **NOIR** sur le folio : cliquez sur le fond du folio pour ajouter un commentaire

Afin de ne pas « masquer » le type des variables, seuls les noms normalisés seront utilisés dans les exemples de cette partie apprentissage **sans créer de table de symboles** : ainsi la première lettre d'une variable rappelle son type. Pour les 5 paragraphes suivants, effectuez en pratique dans Automgen tous les exemples demandés, puis observez leur fonctionnement dans le but de **mémoriser** les principes nouveaux et de les **reproduire** dans la partie application.

I - 2 - Utilisation de nouveaux opérateurs dans les réceptivités et dans les actions

I - 2 - 1 - Les opérateurs utilisables dans une réceptivité :

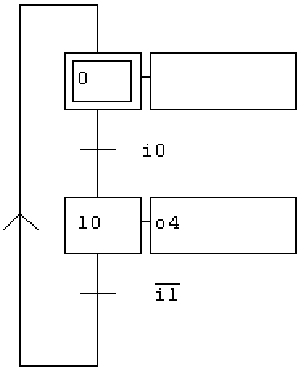
En utilisant l'assistant réalisez le grafcet ci-contre, lancez la simulation, agissez sur les entrées i_0 et i_1 tout en observant les conditions d'activation des étapes et l'évolution du grafcet, puis répondez aux questions suivantes :

A quelle condition la réceptivité i_0 est-elle vraie ? si $i_0=0$ si $i_0=1$

A quelle condition la réceptivité $/i_1$ est-elle vraie ? si $i_1=0$ si $i_1=1$

Est-il possible que toutes les réceptivités de ce grafcet restent vraies simultanément ?

- OUI : expliquez alors à quelle condition :
- NON : expliquez alors pourquoi :



On appelle **front montant** l'instant où une entrée passe du niveau bas au niveau haut [passage de 0 à 1], et **front descendant** l'instant où une entrée passe du niveau haut au niveau bas [passage de 1 à 0]. Sachant que les opérateurs \uparrow et \downarrow utilisés dans les réceptivités s'obtiennent en utilisant les touches « haut » et « bas » du clavier [symbolisées par \uparrow et \downarrow sur le clavier], réalisez le grafcet ci-contre puis répondez aux questions suivantes :

A quelle condition la réceptivité $\uparrow i_0$ est-elle vraie ?

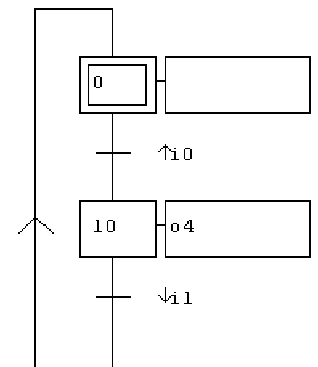
- lors d'un **front montant** sur i_0 lors d'un **front descendant** sur i_0

A quelle condition la réceptivité $\downarrow i_1$ est-elle vraie ?

- lors d'un **front montant** sur i_1 lors d'un **front descendant** sur i_1

Est-il possible que toutes les réceptivités de ce grafcet restent vraies simultanément ?

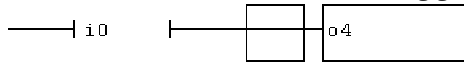
- OUI : expliquez alors à quelle condition :
- NON : expliquez alors pourquoi :



Conclusion : dans une réceptivité on peut tester une entrée sur niveau [détection du *niveau haut* ou du *niveau bas*] ou encore sur front [détection du *front montant* ou du *front descendant*] en utilisant les opérateurs $/$, \uparrow et \downarrow .

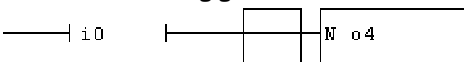
I - 2 - 2 - Les opérateurs utilisables dans une action :

Dans un nouveau folio réalisez le logigramme suivant puis répondez aux questions posées :



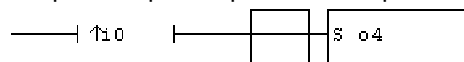
- Quel est l'état logique de o_4 lorsque $i_0=0$? 0 1
- Quel est l'état logique de o_4 lorsque $i_0=1$? 0 1

Modifiez votre logigramme comme ci-dessous en ajoutant l'opérateur \neg dans l'action puis répondez aux questions :

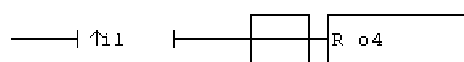


- Quel est l'état logique de o_4 lorsque $i_0=0$? 0 1
- Quel est l'état logique de o_4 lorsque $i_0=1$? 0 1

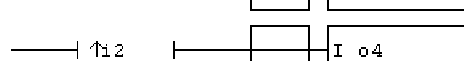
Réalisez maintenant le logigramme suivant utilisant les opérateurs S , R et I dans les actions et l'opérateur \uparrow dans les réceptivités puis répondez aux questions posées :



- Que se passe-t-il à chaque impulsion sur i_0 ?
- o_4 est forcée à 0 o_4 est forcée à 1 o_4 change d'état



- Que se passe-t-il à chaque impulsion sur i_1 ?
- o_4 est forcé à 0 o_4 est forcée à 1 o_4 change d'état



- Que se passe-t-il à chaque impulsion sur i_2 ?
- o_4 est forcée à 0 o_4 est forcée à 1 o_4 change d'état

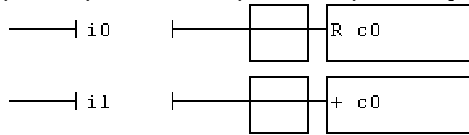
Reliez par un trait chacune des 5 questions suivantes à sa bonne réponse et **mémorisez** à jamais ces 5 remarques :

- | | |
|--|--|
| Quel est le rôle de l'opérateur R dans une action ? ● | ● Si l'étape est active les sorties présentes sont à 0 |
| Quel est le rôle de l'opérateur S dans une action ? ● | ● Bascule la sortie à 1 quelque soit son état |
| Quel est le rôle de l'opérateur \neg dans une action ? ● | ● Change systématiquement l'état de la sortie |
| Quel est le rôle de l'opérateur I dans une action ? ● | ● Si l'étape est active les sorties présentes sont à 1 |
| Comment réagit une action utilisant aucun opérateur ? ● | ● Bascule la sortie à 0 quelque soit son état |

Conclusion : dans une action on peut contrôler l'état d'une sortie ou la manière dont elle bascule en utilisant les opérateurs R , S , I et \neg . Tous ces nouveaux opérateurs sont utilisables dans un logigramme comme dans un grafcet.

I - 3 - Utilisation d'un compteur

Réalisez le logigramme suivant utilisant le compteur $c0$ et les opérateurs R et $+$ dans les actions, lancez la simulation puis répondez aux questions posées (on appelle « impulsion » le passage de 0 à 1 suivi d'un passage de 1 à 0) :



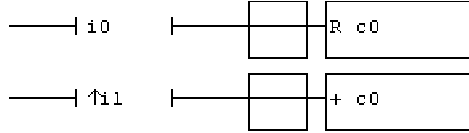
Quel est l'effet d'une impulsion sur $i0$?

- le compteur $c0$ est remis à zéro le compteur $c0$ compte

Que se passe-t-il lorsque $i1$ reste à 1 ?

- $c0$ s'incrémente de 1 unité $c0$ compte en continu

Modifiez votre logigramme en ajoutant l'opérateur \uparrow comme ci-dessous puis répondez aux questions posées :



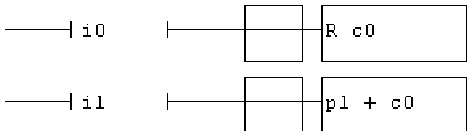
Quel est l'effet d'un front montant sur $i1$?

- $c0$ s'incrémente de 1 unité $c0$ compte en continu

Que se passe-t-il lorsque $i1$ reste à 1 ?

- $c0$ reste constant $c0$ compte en continu

Modifiez votre logigramme en ajoutant l'opérateur $P1$ comme ci-dessous puis répondez aux questions posées :



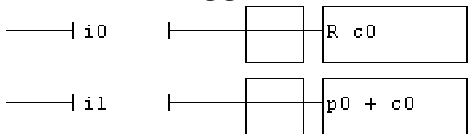
A quel moment le compteur $c0$ s'incrémente-t-il ?

- à l'activation de l'étape à la désactivation de l'étape

Que se passe-t-il lorsque $i1$ reste à 1 ?

- $c0$ reste constant $c0$ compte en continu

Modifiez votre logigramme en utilisant l'opérateur $P0$ comme ci-dessous puis répondez aux questions posées :



A quel moment le compteur $c0$ s'incrémente-t-il ?

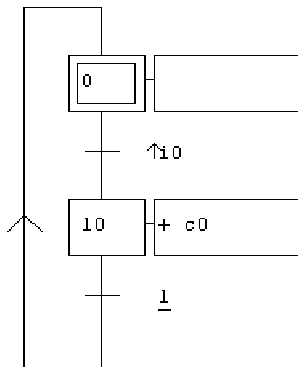
- à l'activation de l'étape à la désactivation de l'étape

Que se passe-t-il lorsque $i1$ reste à 1 ?

- $c0$ reste constant $c0$ compte en continu

Videz votre folio, réalisez le grafctet ci-dessous en utilisant l'assistant puis répondez aux questions posées :

Remarque : la seconde réceptivité marquée 1 est une réceptivité **toujours vraie** et se rédige en écrivant =1 dans la transition.



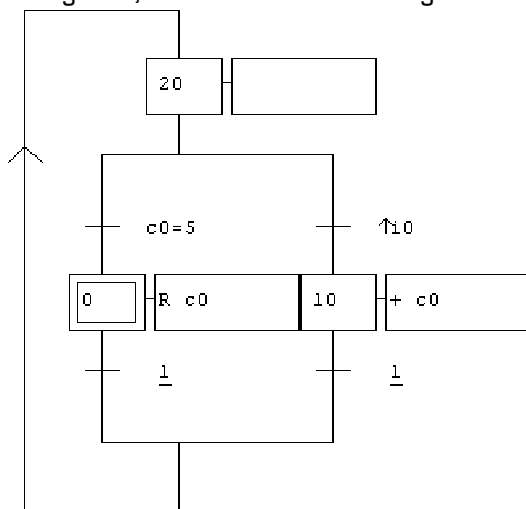
Quelle est la valeur du compteur $c0$ au démarrage ?

Que se passe-t-il lorsque à chaque front montant sur $i0$?

- $c0$ reste constant $c0$ s'incrémente

Le problème de ce grafctet est que le compteur n'est pas mis à zéro au démarrage : il contient une valeur quelconque à la mise sous tension du système. Pour garantir un compteur à zéro, il faudrait mettre l'action $R\ c0$ dans l'étape initiale, mais dans ce cas il serait mis à zéro systématiquement à chaque front montant sur $i0$. La solution pour mettre à zéro un compteur une seule fois au démarrage est donnée ci-dessous.

On utilise pour cela une divergence en OU, mais **l'étape initiale est placée dans une des branches** de la divergence, et non au sommet du grafctet. Créez et testez le grafctet ci-dessous puis répondez aux questions posées :



Quelle est la valeur du compteur $c0$ au démarrage ?

Que se passe-t-il à chaque front montant sur $i0$?

- $c0$ reste constant $c0$ s'incrémente

Quelle est la plus grande valeur que peut prendre le compteur $c0$?

Que se passe-t-il lorsque $c0$ atteint la valeur 5 ?

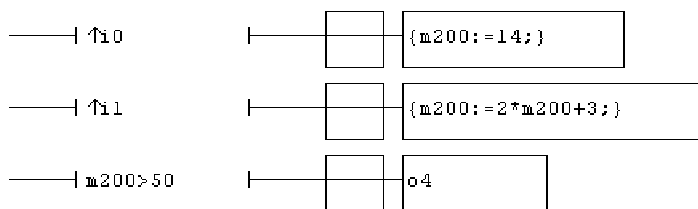
- $c0$ reste à 5 $c0$ est remis à 0

Cet exemple vous a notamment montré :

- * Comment forcer un compteur à 0 au démarrage [opérateur R]
- * Comment tester la valeur d'un compteur [opérateur =]
- * Comment incrémenter un compteur [opérateur +]
- * Comment tester un front montant sur une entrée [opérateur \uparrow]
- * L'utilité d'une réceptivité toujours vraie [notée =1]

I - 4 - Utilisation d'une variable utilisateur

Les variables utilisateur permettent de mémoriser une valeur numérique et d'effectuer des calculs simples dans votre programme (un logigramme ou un grafctet). Les variables utilisateur sont nommées $m200$ à $m9999$ dans Automgen. Réalisez et testez le logigramme suivant utilisant la variable utilisateur $m200$ puis répondez aux questions posées :



Quel est l'effet d'un front montant sur i0 ?

- la variable m200 prend la valeur 14
- la variable m200 prend la valeur 2xm200+3

Quel est l'effet d'un front montant sur i1 ?

- la variable m200 prend la valeur 14
- la variable m200 prend la valeur 2xm200+3

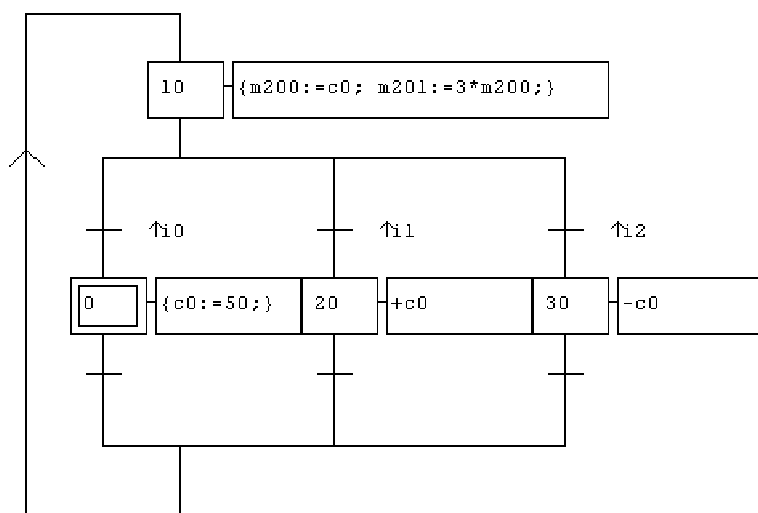
Dans le logigramme ci-dessus à quelle condition la sortie o4 est-elle activée ?

- si m200=50 [égal à]
- si m200<>50 [différent de]
- si m200<=50 [inférieur ou égal]
- si m200>50 [strictement supérieur]

Remarques concernant l'utilisation des variables utilisateur dans Automgen :

- * Les variables utilisateur m200 à m9999 ne peuvent contenir que **des nombres entiers**
- * Les opérateurs { et } (les accolades) dans une étape permettent d'écrire du **code littéral** dans une action
- * Le code littéral est le langage de bas niveau utilisé en interne par Automgen
- * Pour donner la valeur 58 à la variable m200 il faut utiliser le code littéral {m200:=58;} dans une action
- * Dans le code littéral l'opérateur := signifie « **prend la valeur de** »
- * Dans le code littéral chaque ligne doit **finir par un point-virgule**
- * Pour faire un calcul on peut utiliser les opérateurs + [addition], - [soustraction], * [multiplication] et / [division entière] dans le code littéral. Exemple : {m204:=47-m200/3;}
- * Pour tester la valeur d'une variable utilisateur on peut utiliser **les opérateurs arithmétiques** >, <, =, <>, <= ou >= dans une réceptivité

I - 5 - Visualisation de l'état des variables lors de la mise au point du programme



Réalisez le grafcet ci-contre dans lequel :

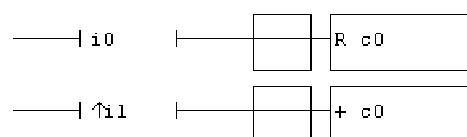
- * Le compteur c0 est initialisé à 50 au démarrage
 - * Un front montant sur i1 **incrémente** c0
 - * Un front montant sur i2 **décrompte** c0
 - * Un front montant sur i0 remet c0 à 50
 - * A tout moment m201 vaut 3 fois la valeur de c0
 - * Le code littéral de l'étape 10 contient 2 lignes
- Lancez la simulation puis agissez sur les entrées i1, i2 et i0 : vous pouvez observer la valeur des variables numériques c0, m200 et m201 directement sur le folio. Pour observer la valeur des variables dans une **dynavue** procédez comme suit :
- * Arrêter la simulation
 - * Dans l'arborescence du projet cliquez droit sur **Mise au point** puis sur **Ouvrir une dynavue**

La **dynavue** se présente dans un nouvel onglet, à côté du folio, et possède un fond rose. Une dynavue permet de visualiser d'un coup toutes les variables du programme sans devoir les rechercher éparpillées dans tout le grafcet. Pour ajouter une nouvelle variable dans la dynavue, cliquez droit sur **Variables** dans la dynavue, cliquez sur **Ajouter**, puis saisissez le nom de la variable [m200 ou i0 par exemple]. Ajoutez les variables i0, i1, i2, c0, m200 et m201 dans votre dynavue, cliquez sur **Mosaïque verticale** dans le menu **Fenêtre** afin d'afficher à l'écran à la fois le folio et la dynavue, lancez la simulation en cliquant sur le bouton **Go !**, puis agissez sur les entrées i0, i1 et i2 en cliquant dans le grafcet tout en observant la valeur des variables dans la dynavue.

Ce paragraphe vous a montré les bases d'utilisation d'une **dynavue** pour visualiser toutes les variables du programme regroupées dans un même document. Automgen dispose d'une seconde solution pour visualiser la valeur et l'évolution des variables durant l'exécution du programme : il s'agit du **monitoring** [disponible également dans l'arborescence **Mise au point**] mais dont le fonctionnement ne sera pas décrit ici. A partir de maintenant vous êtes libre d'utiliser un de ces deux outils disponibles dans Automgen, à partir du moment où ils vous permettent d'organiser la mise au point de votre programme sans perdre trop de temps, que leur utilisation soit demandée ou non dans le sujet des TP.

I - 6 - Affichage de la valeur d'une variable numérique sur un pupitre

Pour visualiser la valeur d'une variable numérique sur un pupitre IRIS 2D il faut y ajouter un nouvel objet appelé « un afficheur ». Les afficheurs sont disponibles dans **Objets prédéfinis** → **Éléments de pupitre** → **Afficheurs** dans l'arborescence des objets IRIS 2D (on a le choix parmi 3 couleurs et 2 tailles différentes). Pour que l'afficheur affiche en temps réel la valeur d'une variable particulière il suffit de renseigner le champ **Variable ou symbole** dans l'onglet **Liens** des propriétés de l'afficheur en y indiquant le nom de la variable à afficher. Dans un nouveau projet, réalisez puis testez le logigramme ci-contre avec un pupitre contenant 2 boutons monostables actionnant les entrées i0 et i1 et un afficheur vert de petite taille affichant la valeur de la variable c0.



I - 7 - Récapitulation de l'ensemble des opérateurs Automgen à connaître

I - 7 - 1 - Les 4 types de réceptivités :

Dans une **réceptivité** il est possible de réaliser 4 types de tests différents comme résumé dans le tableau suivant :

Les 4 types de réceptivités	Exemples
Test d'une entrée [sur niveau ou sur front]	i0 /i2 ↑i5
Test d'une temporisation	t2
Test d'une étape [sur niveau ou sur front]	x4 /x7 ↓x16
Test de la valeur numérique d'une variable ou d'un compteur	m200>8 c0=12

On peut aussi utiliser plusieurs tests dans la même réceptivité à condition de les relier par des **opérateurs logiques**.

I - 7 - 2 - Les 4 types d'actions :

Dans une **action** il est possible d'utiliser 4 types de variables différents comme résumé dans le tableau suivant :

Les 4 types d'actions	Exemples
Activation d'une sortie	o0 Ro4 so9
Déclenchement d'une temporisation	t3(4s)
Incrémementation ou décrémementation d'un compteur	+c0 -c23
Affectation d'une valeur à une variable numérique	{m217:=14;} {m358:=m201+c0;}

On peut aussi réaliser plusieurs actions dans le même rectangle d'action à condition de les séparer par une **virgule**.

I - 7 - 3 - Les opérateurs utilisables dans une réceptivité :

Afin de contrôler la manière dont les entrées ou les variables sont détectées ou testées dans une réceptivité il est possible d'utiliser l'ensemble des opérateurs suivants.

Les opérateurs utilisables dans une réceptivité	Rôle
.	ET logique
+	OU logique
/	NON logique, détecte un niveau bas
↑	Détecte un front montant
↓	Détecte un front descendant
= < <= > >= <>	Compare deux valeurs numériques
=1	1 logique : condition toujours vraie

Une réceptivité se rédige entre les délimiteurs dans un logigramme, ou bien à droite d'une transition dans un grafcet.

I - 7 - 4 - Les opérateurs utilisables dans une action :

Afin de contrôler la manière dont les sorties ou les variables sont activées ou modifiées dans une action il est possible d'utiliser l'ensemble des opérateurs suivants.

Les opérateurs utilisables dans une action	Rôle
,	Séparateur pour réaliser plusieurs actions
R	Reset : met à 0 une sortie ou un compteur
S	Set : met à 1 une sortie
I	Inversion : complémente l'état de la sortie
N	Négation logique : désactive la sortie
P1	Impulsion à l'activation de l'étape
P0	Impulsion à la désactivation de l'étape
+	Incrémente un compteur [y additionne 1]
-	Décrémente un compteur [y soustrait 1]
{ et }	Permet d'écrire du code littéral

Une action se rédige dans un rectangle d'action, aussi bien dans un logigramme que dans un grafcet.

Remarques à surligner et à retenir :

- * L'opérateur . ne s'utilise que dans une réceptivité et **n'a aucun sens dans une action !**
- * L'opérateur , ne s'utilise que dans une action et **n'a aucun sens dans une réceptivité !**
- * L'opérateur + a **deux fonctions différentes** selon qu'il est utilisé dans une action ou dans une réceptivité
- * Pour obtenir la liste exhaustive de tous les opérateurs utilisables consultez la documentation officielle d'Automgen

II - Partie application

En utilisant toutes vos connaissances concernant Automgen, proposez une solution sous forme d'un grafcet associé à un pupitre pour chacune des applications suivantes dans lesquelles seule la problématique à résoudre est indiquée.

II - 1 - Application 1 : le chronomètre

On désire réaliser un chronomètre possédant 3 boutons monostables nommées **START**, **STOP**, et **RAZ** et dont le fonctionnement doit être le suivant :

- * Le chronomètre doit mesurer et afficher le temps à 100 ms près
- * Au démarrage le chronomètre doit être initialisé automatiquement à zéro
- * Une impulsion sur le bouton **START** lance la mesure du temps qui est alors affichée en temps réel
- * Une impulsion sur **STOP** arrête la mesure du temps et fige l'affichage à la valeur mesurée
- * Une impulsion sur **RAZ** [RAZ = Remise **A** Zéro] remet le chronomètre à zéro
- * Le bouton **RAZ** n'a aucun effet si la mesure du temps n'a pas été arrêtée par le bouton **STOP**

II - 1 - 1 - Proposez et validez dans un nouveau projet Automgen une solution composée d'un grafcet associé à un pupitre pour réaliser ce chronomètre. Les trois symboles **START**, **STOP**, et **RAZ** seront créés et connectés respectivement aux entrées i_0 , i_1 et i_2 , et le compteur c_0 sera utilisé pour mesurer le temps.

II - 2 - Application 2 : le compteur de passage

Pour contrôler le nombre de voitures présentes dans un parking souterrain, un dispositif appelé « **compteur de passage** » a été installé. Le compteur de passage permet au gardien du parking de consulter sur un afficheur le nombre exact de voiture présentes dans le parking, et de pouvoir interdire l'accès au parking lorsqu'il est plein :

- * Si une voiture entre dans le parking l'afficheur **s'incrémente**
- * Si une voiture sort du parking l'afficheur **se décrémente**

Pour détecter le sens de passage d'une voiture deux capteurs **A** et **B** ont été placés à l'entrée du parking, et sont espacés de 10 cm :

- * Une voiture qui entre active le capteur **A avant le B** → Sens d'entrée d'une voiture →
- * Une voiture qui sort active le capteur **B avant le A** ← Sens de sortie d'une voiture ←

Dans les deux sens de circulation la voiture passera forcément devant les 2 capteurs à la fois. Sachant qu'un capteur délivre un 1 logique en cas de présence d'une voiture devant lui, et délivre un 0 logique en cas d'absence de voiture, les séquences délivrées par les deux capteurs et permettant de détecter le sens de circulation sont les suivantes :

Séquence délivrée par les capteurs	La voiture approche	Elle passe devant le 1 ^{er} capteur	Elle passe devant les 2 capteurs	Elle passe devant le 2 nd capteur	La voiture s'éloigne
Une voiture entre	A=0 et B=0	A=1 et B=0	A=1 et B=1	A=0 et B=1	A=0 et B=0
Une voiture sort	A=0 et B=0	A=0 et B=1	A=1 et B=1	A=1 et B=0	A=0 et B=0

Le sens de circulation de la voiture est donc déterminé en analysant l'ordre dans lequel les capteurs **A** et **B** sont activés [A, AB, puis B : une voiture entre. B, AB, puis A : une voiture sort].

II - 2 - 1 - Proposez et validez dans un nouveau projet Automgen une solution composée d'un grafcet associé à un pupitre pour réaliser ce compteur de passage. Le pupitre doit contenir 2 boutons bistables **A** et **B** simulant les capteurs à l'entrée du parking et un afficheur indiquant le nombre de voiture présente dans le parking :

- * Les capteurs **A** et **B** seront connectés aux entrées i_0 et i_1 dans la table des symboles
- * Le nombre de voitures sera comptabilisé dans la variable **N** associé au compteur c_0 dans la table des symboles
- * Au démarrage on considère que le parking est vide [**N = 0**]
- * En simulant l'entrée d'une nouvelle voiture avec les boutons **A** et **B** le compteur doit s'incrémenter
- * En simulant la sortie d'une voiture avec les boutons **A** et **B** le compteur doit se décrémente
- * Si **N = 0** [parking vide] alors la sortie d'une voiture ne doit pas modifier la valeur du compteur [qui doit rester à 0]

II - 2 - 2 - Ajoutez 3 voyants **VERT**, **ORANGE** et **ROUGE** sur le pupitre indiquant le remplissage du parking :

- * On considère que le parking a une capacité de 10 places seulement [pour pouvoir le remplir rapidement]
- * Les voyants **VERT**, **ORANGE** et **ROUGE** seront connectés respectivement aux sorties o_0 , o_1 et o_2
- * Le voyant **VERT** doit être allumé lorsque le nombre de places vides est supérieur ou égal à 4
- * Le voyant **ORANGE** doit être allumé lorsque le nombre de places vides est strictement inférieur à 4
- * Le voyant **ROUGE** doit être allumé lorsque le parking est plein [10 voitures présentes, aucune place vide]

II - 2 - 3 - Sachant que chaque voiture entrante paye 2 € pour se garer au parking, ajoutez un second afficheur au pupitre indiquant la somme présente dans la caisse à tout moment de la journée. Cette somme d'argent sera mémorisée dans la variable **CAISSE** associée à la variable utilisateur m_{200} : une voiture entrante incrémente la caisse de 2 €, une voiture sortante n'a aucun effet sur le contenu de la caisse, et au démarrage la caisse est vide.