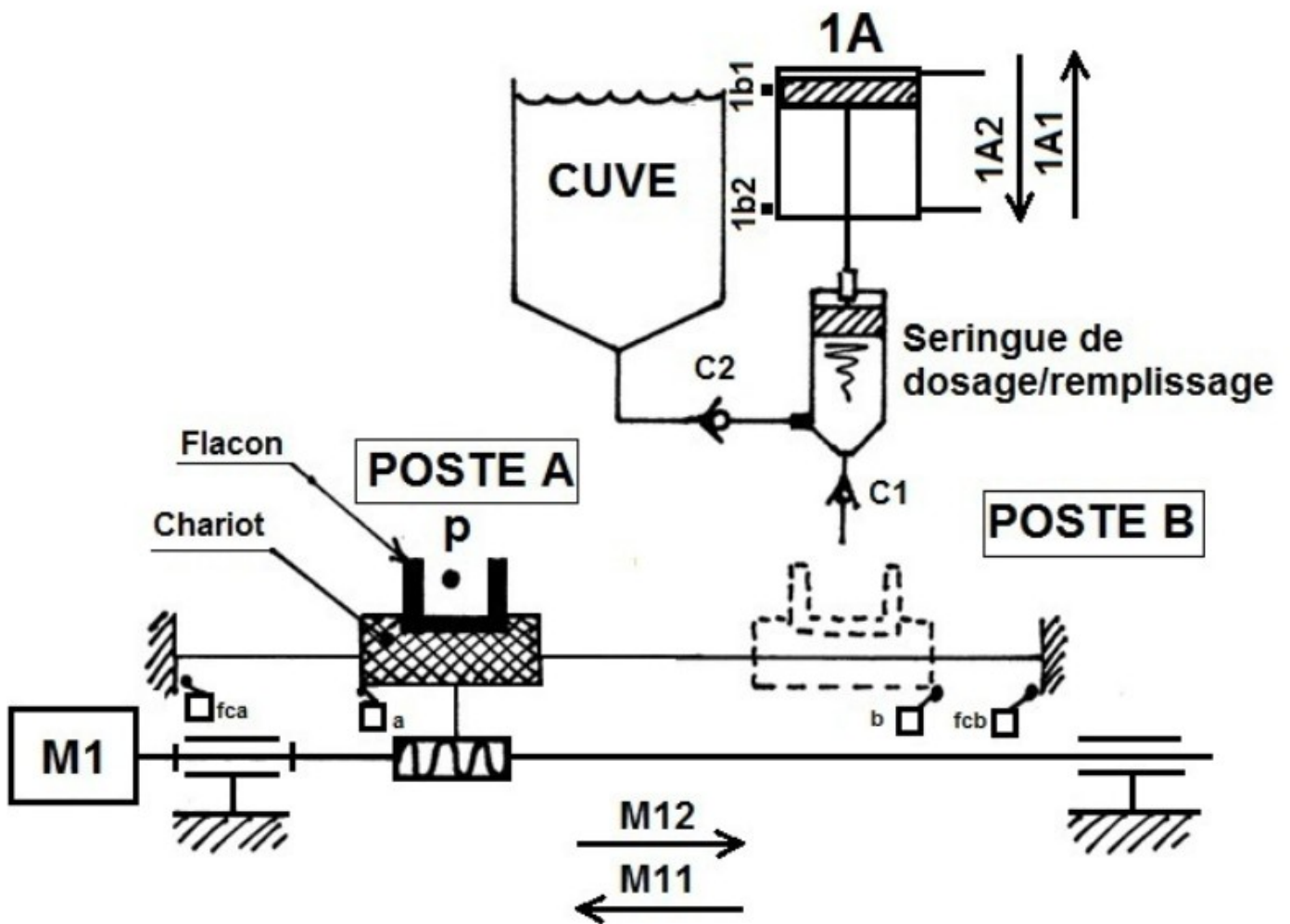


Nom:  
 Prénom:  
 groupe de labo:  
 S2\_labo\_flacon\_ST\_2022P.odt

durée: P1 - 6x2UC  
 2UC/semaine - 2022P

## Séance N°2 et 3: Machine de remplissage de flacons

objectif: Mettre en œuvre les concepts de base du grafcet sur un exemple concret, programmation en langage SFC, LADDER et ST sur UnityPro.



### 1. Présentation du système

#### 1.1 Système de transfert

Un chariot permet d'assurer le transfert du flacon entre les deux postes. Le chariot est guidé en liaison glissière par rapport au bâti et son déplacement est assuré par un système vis/écrou entraîné par le moteur à courant continu **M1**. Les deux capteurs à action mécanique **a** et **b** détectent les positions du chariot correspondant aux postes **A** et **B**. Les deux capteurs de fin de courses **fca** et **fcb** détectent les butées mécaniques de fin de course de la glissière. Un capteur photo-électrique **p** (fixé au bâti), permet de détecter la présence d'un flacon sur le chariot placé au poste **A**.

- poste **A** : poste de chargement/déchargement du flacon. (non étudié)
- poste **B** : poste de remplissage.

## 1.2 Poste B

Le poste **B** est constitué d'une cuve de stockage du produit destiné au remplissage du flacon. Une seringue de dosage/remplissage, entraînée par la tige du vérin double effet **1A**, permet l'aspiration du produit et son injection dans le flacon. Le clapet **C1** permet d'éviter l'aspiration d'air dans la phase de dosage (aspiration du produit).

Le clapet **C2** permet d'éviter le refoulement du produit dans la cuve dans la phase de remplissage du flacon. La tige du vérin **1A** est liée mécaniquement au piston de la seringue de dosage/remplissage. Deux capteurs **1b1** et **1b2** détectent les positions de fin de course du piston.

## 1.3 Description du pupitre opérateur

nom	type	rôle
<b>Sm</b>	Bpm NO	mise en énergie
<b>Sa</b>	Bpm NF	arrêt énergie
<b>dcy</b>	Bpm NO	départ cycle
<b>acq</b>	Bpm NO	acquiescement opérateur
<b>au</b>	Bcdp NF	arrêt d'urgence à accrochage (déverrouillage par quart de tour)

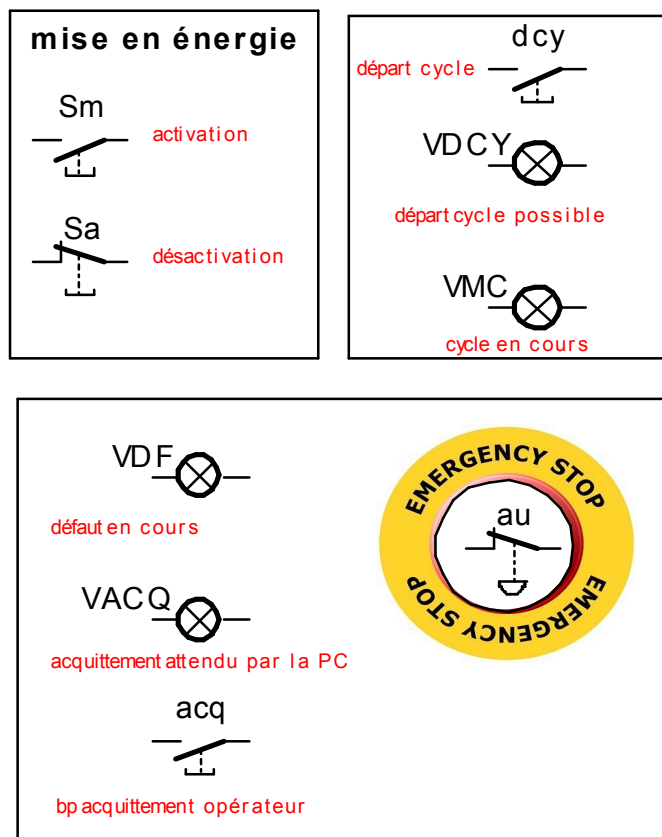
*Bpm* : bouton poussoir monostable

*Bcdp* : bouton « coup de poing »

*NO* : normalement ouvert

*NF* : normalement fermé

### PUPITRE OPERATEUR



nom	type	rôle	Conditions d'allumage
<b>VDCY</b>	voyant	départ cycle possible	Clignotement <b>1Hz</b> si toutes les conditions nécessaires au lancement d'un cycle sont réunies avant l'appui sur dcy.
<b>VMC</b>	voyant	cycle en cours	Allumage <b>continu</b> si le cycle automatique est en cours.
<b>VACQ</b>	voyant	Acquiescement attendu par la partie commande	Clignotement <b>1Hz</b> lorsque la partie commande attend un acquiescement opérateur, suite à une action de sa part.
<b>VDF</b>	voyant	Défaut en cours de traitement	Clignotement [2 éclats brefs de <b>0,2s</b> + 1 éclat long de <b>0,8s</b> avec <b>0,2s</b> éteint entre chaque éclat] si un défaut est en cours. Un défaut est en cours lorsque le relais de sécurité <b>KAS</b> est désactivé <u>ou</u> lorsque le temps enveloppe du <u>dernier mouvement réalisé du chariot du poste A au poste B</u> n'a pas été respecté.

## 1.4 Cycle automatique de remplissage

### 1.4.1 Position initiale de la PO

Au démarrage du cycle,

- le chariot est au poste **A**,
- un flacon vide y est présent,
- la seringue est remplie de produit.

### 1.4.2 Condition de lancement de cycle

Le cycle est lancé si :

- l'opérateur le demande par **un appui prolongé de 2s** sur **dcy**,
- les conditions initiales du cycle liées à la PO sont respectées.

### 1.4.3 Description du cycle

Quand le cycle est lancé, les opérations suivantes se succèdent **N=3** fois:

- avant le démarrage du cycle : initialisation du compteur : **N=0**
- le chariot se déplace au poste **B**,
- une attente de **3s** permet la stabilisation mécanique du chariot
- le flacon est rempli par la sortie du vérin de remplissage **1A**,
- une attente de **3s** permet d'attendre la fin de l'écoulement du produit
- 2 actions simultanées sont ensuite réalisées :
  - # une dose de produit est préparée dans la seringue par le retour du vérin **1A** en position rentrée,
  - # le chariot revient au poste **A**.
- l'opérateur place un flacon vide sur le chariot et acquitte son intervention sur le pupitre opérateur.
- incrémentation du compteur **N=N+1**

Le cycle recommence jusqu'à **N=3**, puis s'arrête pour attendre le prochain départ cycle.

## 1.5 Surveillance du temps enveloppe

Une action commandée dans la partie opérative (PO) dure « un certain temps » qui peut fluctuer(\*) dans une tolérance que l'on appelle le **temps enveloppe**.











(\*) chute de pression (fuites?), fluctuation de la charge de l'actionneur, modification des frottements, blocage mécanique...

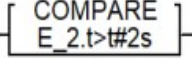





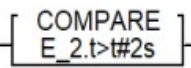









- Si le **temps réel de l'action** de la PO dépasse le temps enveloppe, une variable défaut [**ERR\_tps\_MAX**] doit être générée.
- Si le **temps réel de l'action** de la PO n'atteint pas le temps enveloppe, une variable défaut [**ERR\_tps\_min**] doit être générée. (généralement dans ce cas, il s'agit d'un défaut capteur : le capteur de fin de mouvement est déjà activé au début de la commande (blocage mécanique du capteur par ex) => l'étape de commande dans le grafset est alors instable, donc son temps d'activation ne dépasse pas le temps de traitement du programme soit environ 50ms)

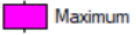












Dans notre thème, la surveillance est limitée à un seul mouvement, mais industriellement, chaque action dans la partie opérative est surveillée par un temps enveloppe.

**Attention : la remise à zéro de la variable [défaut temps enveloppe] ne doit être réalisée que lorsque le mouvement a été refait au moins une fois à l'intérieur du temps enveloppe.**

## 2. Travail demandé

	<b>Légende des symboles</b>		
	Travail de lecture de documents explicatifs		
	Travail de mise en œuvre/câblage/test de la partie opérative		
	Programmation sur UnityPro		
	Travail nécessitant un transfert de programme et une connexion UnityPro <=> API		
1	<p>Réaliser le <b>câblage de la partie opérative</b> :</p> <ul style="list-style-type: none"> <li>- circuit de puissance pneumatique (vérin 1A)</li> <li>- circuit de sécurités câblées (relais de sécurité KAS et 0YV12_bis)</li> <li>- circuit de puissance électrique (moteur M1)</li> </ul> <p> bien respecter l'adressage !</p>		
2	<p><b>Tester</b> le bon fonctionnement de votre câblage(*) à l'aide des tables d'animation et en manoeuvrant l'arrêt d'urgence pour tester l'activation/désactivation du relais de sécurité <b>KAS</b>.</p> <p>(*) capteurs, distributeurs, actionneurs, relais de sécurité, voyants, boutons/sélecteurs du pupitre...</p>		
3	<p><b>Lire</b> dans le manuel, les explications concernant l'utilisation des blocs fonctions [COMPARE, OPERATE, TON (temporisation), CTU (compteur)]</p> <p><b>5 BLOCS FONCTIONS</b></p> <ol style="list-style-type: none"> <li>1. Types de Blocs fonctions utilisés dans Unity . . . . .</li> <li>2. Bloc <b>COMPARE</b> . . . . .</li> <li>3. Bloc <b>OPERATE</b> . . . . .</li> <li>4. Création de la section <i>Blocs_Fonctions_Var</i> dans le Navigateur du Projet . . . . .</li> <li>5. Bloc fonction <b>TON</b> . . . . .       <ol style="list-style-type: none"> <li>5.1 MÉTHODE 1 : Création du Temporisateur TON à partir de la sélection de données . . . . .</li> <li>5.2 MÉTHODE 2 : Création du Temporisateur TON à partir de l'assistant de saisie FFB . . . . .</li> <li>5.3 Sélection de la réceptivité liée à la temporisation dans la section CHART . . . . .</li> </ol> </li> <li>6. Bloc Fonction <b>CTU</b> . . . . .       <ol style="list-style-type: none"> <li>6.1 Création du Compteur CTU à partir de l'assistant de saisie FFB . . . . .</li> <li>6.2 Sélection de la réceptivité liée au compteur dans la section CHART . . . . .</li> </ol> </li> </ol> <p>Lire les consignes de programmation à suivre scrupuleusement :</p> <p><b>20 CHECK UP programme</b></p> <ol style="list-style-type: none"> <li>1. Dépôt de programme sous moodle . . . . .</li> <li>2. Jetons . . . . .</li> <li>3. Commandes utiles . . . . .       <ol style="list-style-type: none"> <li>3.1 Insertion ou suppression de lignes en ladder . . . . .</li> <li>3.2 Réinitialisation . . . . .</li> <li>3.3 Accès à des informations . . . . .</li> </ol> </li> <li>4. Liste non exhaustive ci-dessous des consignes à suivre . . . . .</li> </ol>		
4	<p><b>Télécharger</b> sur moodle le programme de départ en .ZEF il contient la configuration, les variables et la section de fusion des variables.</p>		

4	<p><b>Réaliser la programmation</b> sur UnityPro du grafcet <b>G0 : AUTO</b></p> <ul style="list-style-type: none"> <li>- la temporisation <b>2s/dcy</b> associée à la transition (<b>t0</b>) sera programmée avec un bloc fonction <b>TON</b>.</li> <li>- les autres temporisations seront programmées avec des blocs fonctions COMPARE utilisant les mots temps d'activité d'étapes <b>[nom d'étape].T</b> </li> <li>- le compteur <b>C</b> sera intégralement programmé avec un bloc fonction <b>CTU</b></li> <li>- les actions mémorisées seront programmées avec des bobines <b>SET</b> et <b>RESET</b> : <math>\neg(S)</math></li> <li>- le déclenchement des actions mémorisées à l'activation ou à la désactivation des étapes sera programmé par des fronts montants/descendants : <math>\neg P </math></li> <li>- Le clignotement des voyants ne sera pas programmé à ce stade.</li> </ul> <p> bien commenter votre programme !</p>		
5	<p><b>Tester</b> le fonctionnement de la partie opérative :</p> <ul style="list-style-type: none"> <li>- vérifier que le cycle des actionneurs obtenu respecte bien le cahier des charges</li> <li>- visualiser l'état des variables sur vos tables d'animations (E/S, temporisations et compteur C)</li> <li>- vérifier le bon fonctionnement des voyants en fonction des différentes phases du cycle (état initial / cycle en cours / attente d'acquiescement...)</li> </ul>		
6	<p><b>Demander une validation</b> du professeur avant de continuer</p>		
7	<p><b>Programmer</b> les grafquets de clignotement :</p> <ul style="list-style-type: none"> <li>- <b>G10 : CLIGN_101010</b> avec des blocs fonctions TON</li> <li>- <b>G11 : CLIGN_101020</b> avec des blocs fonctions COMPARE utilisant les mots temps d'activité d'étapes <b>[nom d'étape].T</b> </li> </ul>		
8	<p><b>Tester</b> le fonctionnement de la partie opérative :</p> <ul style="list-style-type: none"> <li>- bien vérifier les périodes de clignotement des voyants, corriger toute instabilité !</li> </ul>		
9	<p><b>Programmer</b> les grafquets <b>G2 : SURV_X1_KM12</b> et <b>G1 : VISU</b></p> <ul style="list-style-type: none"> <li>- les temporisations seront programmées avec des blocs fonctions COMPARE utilisant les mots temps d'activité d'étapes <b>[nom d'étape].T</b> </li> </ul>		
10	<p><b>Tester</b> le fonctionnement du voyant <b>VDF</b>:</p> <ul style="list-style-type: none"> <li>- allumage de <b>VDF</b> en fonction de l'état du relais de sécurité <b>KAS</b></li> <li>- temps enveloppe min (*) de l'action <b>KM12</b> associée à l'étape 1</li> <li>- temps enveloppe MAX (*) de l'action <b>KM12</b> associée à l'étape 1</li> <li>(*) faire varier le temps d'activation de l'étape 1 en manoeuvrant le capteur <b>a</b> sur le pupitre.</li> </ul>		
11	<p><b>Demander une validation</b> du professeur avant de continuer</p>		
12	<p><b>Lire</b> dans le manuel les explications sur les variables <b>nom_étape.tminErr</b> et <b>nom_étape.tmaxErr</b> associées au étapes du diagramme SFC :</p>		

	<b>12 CRÉATION DU GRAPHE D'ÉTAT</b> 1. Création de la structure (langage SFC) . . . . . 2. Commentaires . . . . . 3. Explication sur les temps enveloppes sur un exemple quelconque . . . . .		
13	<b>Modifier</b> le programme en supprimant le grafctet <b>G2 : SURV_X1_KM12</b> et en utilisant les variables <b>nom_étape.tminErr</b> et <b>nom_étape.tmaxErr</b> pour générer les variables <b>ERR_tps_min</b> et <b>ERR_tps_max</b> utilisées dans les conditions d'allumage du voyant VDF.  Maximum t#10s  Minimum t#5s  la remise à zéro des variables <b>ERR_tps_min</b> ou <b>ERR_tps_max</b> ne doit être réalisée que lorsque le mouvement a été refait <u>au moins une fois</u> à l'intérieur du temps enveloppe.		
14	<b>Tester</b> le fonctionnement du voyant <b>VDF</b> : - temps enveloppe min (*) de l'action <b>KM12</b> associée à l'étape 1 - temps enveloppe MAX (*) de l'action <b>KM12</b> associée à l'étape 1 (*) faire varier le temps d'activation de l'étape 1 en manoeuvrant le capteur <b>a</b> sur le pupitre.		
15	<b>Demander une validation</b> du professeur avant de continuer		
16	<b>Découvrir</b> et mettre en oeuvre les ressources proposées sur moodle sur la transcription d'un grafctet en langage ST.  <b>RESSOURCES:</b>  <b># LANGAGE ST</b> <ul style="list-style-type: none"> <li>transcription d'un grafctet en programme littéral structuré (langage ST)            Cette ressources vous expliquera les étapes et les difficultés à surmonter pour transcrire un grafctet simple en programme littéral structuré (langage ST), deux fichiers fluidsims (1 et 2) vous permettront de comprendre la notion de perte d'activité d'étape en utilisant vos connaissances de S1.</li> <li>le programme UnityPro en langage ST obtenu vous permettra de manipuler ce nouveau langage sur un exemple. Le format du programme est en .ZEF pour permettre l'ouverture sur les différentes versions d'UnityPro.</li> <li>ces explications sur la programmation et le test en mode RUN avec le langage ST vous apprendront à utiliser les outils de de visualisation de l'état des variables pour "debugage" du programme.</li> </ul>		
17	<b>Programmer</b> le grafctet <b>G0:AUTO</b> en langage ST		
18	<b>Tester</b> le fonctionnement complet du programme avec cette modification		
19	<b>Programmer</b> des écrans d'exploitation permettant le test du programme en mode simulation.		
20	<b>Demander une validation finale</b> du professeur		

<b>D*</b>	<b>explications</b>	<b>technologie</b>	<b>Symb UP*</b>	<b>Pupitre *</b>	<b>Banc *</b>	<b>Adresse*</b>
<b>ENTREES API</b>						
<b>1b1</b>	Vérin 1A rentré	Action mécanique NO	_1b1		x	%I0.2.1
<b>1b2</b>	Vérin 1A sortie	Action mécanique NO	_1b2		x	%I0.2.2
<b>p</b>	Présence pièce poste A	Sélecteur bi-stable	p	X <sub>par</sub>		%I0.2.0
<b>a</b>	Chariot au poste A	Sélecteur bi-stable	a	X <sub>f4</sub>		%I0.2.40
<b>b</b>	Chariot au poste B	Sélecteur bi-stable	b	X <sub>f3</sub>		%I0.2.39
<b>kas</b>	Relais de sécurité	Contact auxiliaire NO	kas		x	%I0.2.9
<b>dcy</b>	départ cycle	Bpm NO	dcy	X <sub>dcy</sub>		%I0.2.32
<b>acq</b>	acquiescement opérateur	Bpm NO	acq	X <sub>init</sub>		%I0.2.33
<b>SORTIES API</b>						
<b>1YV14</b>	SORTIR vérin 1A : REMPLISSAGE	Bobine de Cd distributeur	_1YV14		x	%Q0.3.1
<b>1YV12</b>	RENTREER vérin 1A : DOSAGE	Bobine de Cd distributeur	_1YV12		x	%Q0.3.2
<b>KM11</b>	DEPLACER CHARIOT au poste A	Bobine de Cd relais	KM11		x	%Q0.3.3
<b>KM12</b>	DEPLACER CHARIOT au poste B	Bobine de Cd relais	KM12		x	%Q0.3.4
<b>VDCY</b>	départ cycle possible	voyant	VDCY	X <sub>VRM</sub>		%Q0.3.5
<b>VMC</b>	cycle en cours	voyant	VMC	X <sub>VMC</sub>		%Q0.3.19
<b>VACQ</b>	Acquiescement attendu par la partie commande	voyant	VACQ	X <sub>VACQ-VINIT</sub>		%Q0.3.25
<b>VDF</b>	Défaut en cours de traitement	voyant	VDF	X <sub>VDF</sub>		%Q0.3.20

**D** : désignation

**Symb UP** : symbole à utiliser dans le programme UnityPro

**Pupitre** : composants déjà câblés sur le pupitre

**Banc** : composants à câbler sur le banc

**Adresse** : adresse des variables sur les cartes d'entrées/sorties de l'automate