

## Examen d'Automatismes

**Enseignant :** TOGUYENI Armand

**Temps imparti :** 2 heures    **Documents :** autorisés

**Calculatrice :** autorisée

### Exercice 1 : Automatisation d'une écluse

Les Hauts de France comptant de nombreux canaux et donc d'écluse, l'entreprise GROSSOUS a été chargée sélectionnée par le Conseil Régional pour l'automatisation de ces écluses.

Une écluse (**Figure 1**) est composée de deux portes encadrant un SAS (station 3) entre le bief amont (station 1) et le bief aval (station 2). Pour franchir le dénivelé entre l'amont et l'aval, un navire se place dans le SAS. Il attend ensuite que l'entrée d'eau de l'amont, ou la sortie de l'eau en aval, le place au niveau de l'eau du bief suivant. La porte correspondante de l'écluse s'ouvre alors, libérant le navire qui peut poursuivre sa navigation.



Figure 1. Photo d'une écluse

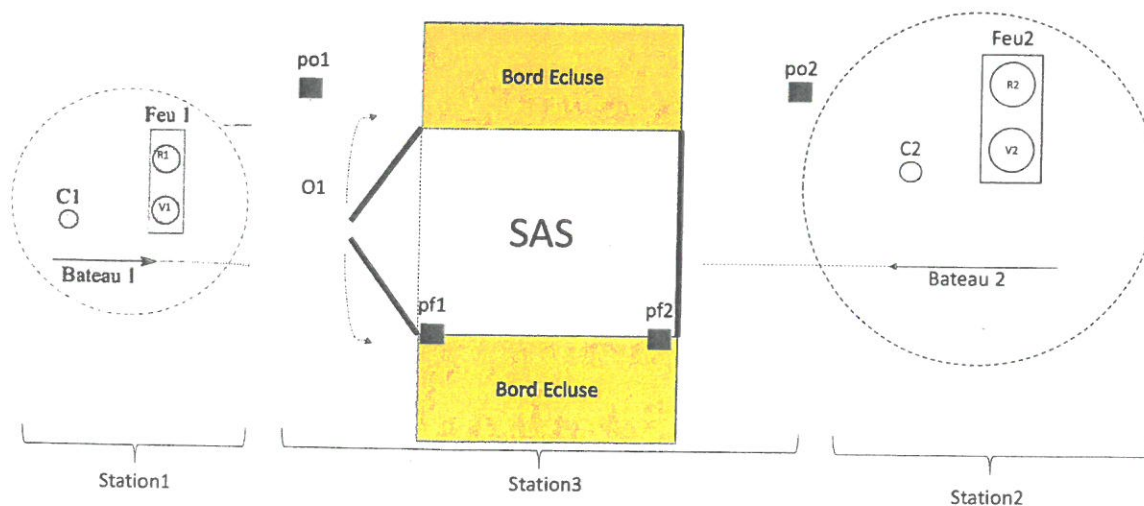


Figure 2. Schéma de l'écluse avec son instrumentation

Nous considérons le système représenté par les Figure 2 et Figure 3. La station 1 est composée du capteur **c1** qui sert à détecter l'arrivée d'un bateau voulant emprunter le SAS et du feu1. Le feu1 est rouge si **R1** vaut 1 et il est vert si **V1** vaut 1. Le rouge indique l'interdiction au bateau d'avancer vers le SAS. Le feu est positionné au vert uniquement si le SAS est vide et que le niveau d'eau dans le SAS est le niveau haut, et que les portes de l'écluse en amont sont ouvertes.

La station 2 est composée du capteur **c2** et du feu2. Elle fonctionne de manière analogue à la station 1. Son feu passe au vert uniquement si l'écluse est vide est que son niveau d'eau est bas, et que les portes avals (côté station 2) sont ouvertes.

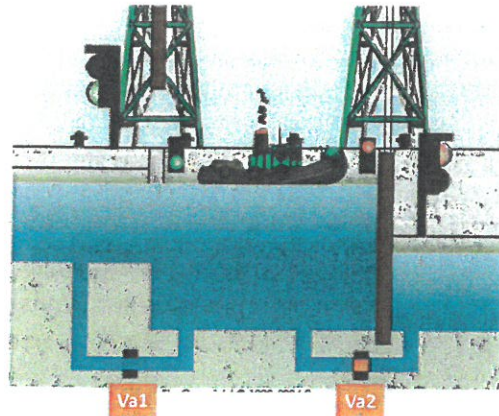


Figure 3. Vue de en coupe de l'écluse avec les vannes Va1 et Va2 utiliser pour la commande des niveaux d'eau dans le sas.

Station	Capteur/Actionneur	Fonction
API Station 1	c1	Détection de l'arrivé d'un bateau avant le SAS au niveau de la station 1
	R1	Allumer le feu rouge de la station 1
	V1	Allumer le feu vert de la station 1
API Station 3	c3	Détection de l'arrivé d'un bateau avant le SAS au niveau de la station 3
	R3	Allumer le feu rouge de la station 3
	V3	Allumer le feu vert de la station 3
API Station 2	Va1	Baisser le niveau d'eau du SAS
	Va2	Monter le niveau d'eau du SAS
	cva1	Capteur vanne Va1 ouverte
	cva2	Capteur vanne Va2 ouverte
	h2	Capteur niveau d'eau haut
	b2	Capteur niveau d'eau bas
	O1	ouverture porte du SAS côté station 1
	O2	ouverture porte du SAS côté station 2
	c2	présence d'un bateau dans le SAS
	po1	portes du SAS côté station 1 ouvertes
	pf1	portes du SAS côté station 1 fermées
	po2	portes du SAS côté station 2 ouvertes
	pf2	portes du SAS côté station 2 fermées

Tableau 1. Liste des capteurs/actionneurs de chaque station

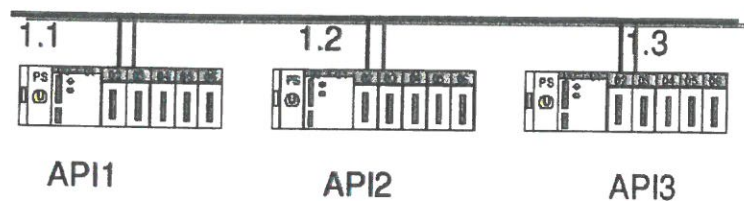


La station 3 correspond au SAS de l'écluse. De part et d'autre, le SAS peut être isolé par des portes (cf. Figure 1). Les portes côté amont (station1) sont ouvertes par la commande **O1**. Cette commande doit être maintenue pour que la porte reste ouverte. Quand les portes sont ouvertes, le capteur **po1** est à 1. Quand elles sont fermées le capteur **pf1** est à 1. Les portes côté aval (station 2) fonctionnent de manière analogue.

Le niveau d'eau dans l'écluse est réglé à l'aide des vannes **Va1** et **Va2** (Figure 3), les portes de l'écluse étant fermées. L'ouverture de **Va1** et la fermeture de **Va2** permet de remplir le SAS. Inversement la fermeture de **Va1** et l'ouverture de **Va2** permet de vider le SAS et donc de baisser le niveau de l'eau au niveau de celui de l'aval (station 2). Les vannes sont à commandes monostables.

L'ensemble des capteurs et actionneurs des 3 stations sont résumés par le Tableau 1.

1. Modéliser en RdPIC la commande de ce système en supposant que l'ensemble des 3 stations est contrôlé par un seul automate programmable industriel.



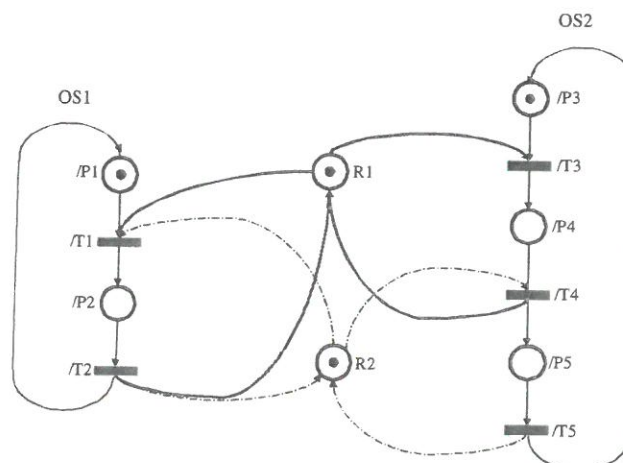
Erreur ! Source du renvoi introuvable. **Figure 4.**  
Architecture opérationnelle pour la mise en œuvre (implantation)

On suppose à présent que chaque station est commandée par son propre automate. La station 1 par l'API1, la station 2 par l'API2 et la station 3 par l'API3 (cf. Figure 4). Les API communiquent par messagerie industrielle sur un réseau industriel de type FIPWAY.

2. Modéliser en RdPIC la commande distribuée de la station 1.
3. Modéliser en RdPIC la commande distribuée de la station 2.
4. Modéliser en RdPIC la commande distribuée de la station 3.

## Exercice 2

On considère le système dont le comportement est modélisé par le réseau de Petri ordinaire de la figure 1.



**Figure 1 : Conception préliminaire de la commande du système considéré**

On désire effectuer une implantation distribuée sur 3 automates programmables industriels de type Premium de chez Schneider (cf. Figure 4). La tâche « OS1 » sera codée sur l'automate API1. La tâche « OS2 » sera codée sur l'automate API2. Les ressources sont gérées par l'automate API3. Les 3 automates (API) communiquent par messagerie industrielle au travers d'un réseau local industriel.

1. Quelle contrainte opérationnelle doit-on prendre en compte pour passer du modèle de spécification de la commande du système donnée par le RdP de la figure à une conception détaillée en RdP ?

Pour établir le modèle de conception détaillé on utilise des sémaphores (Figure 5)

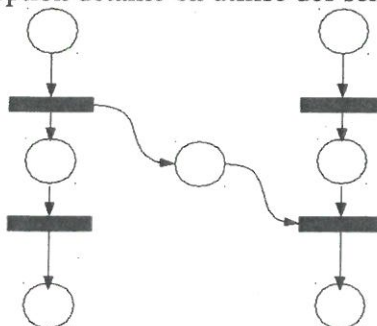
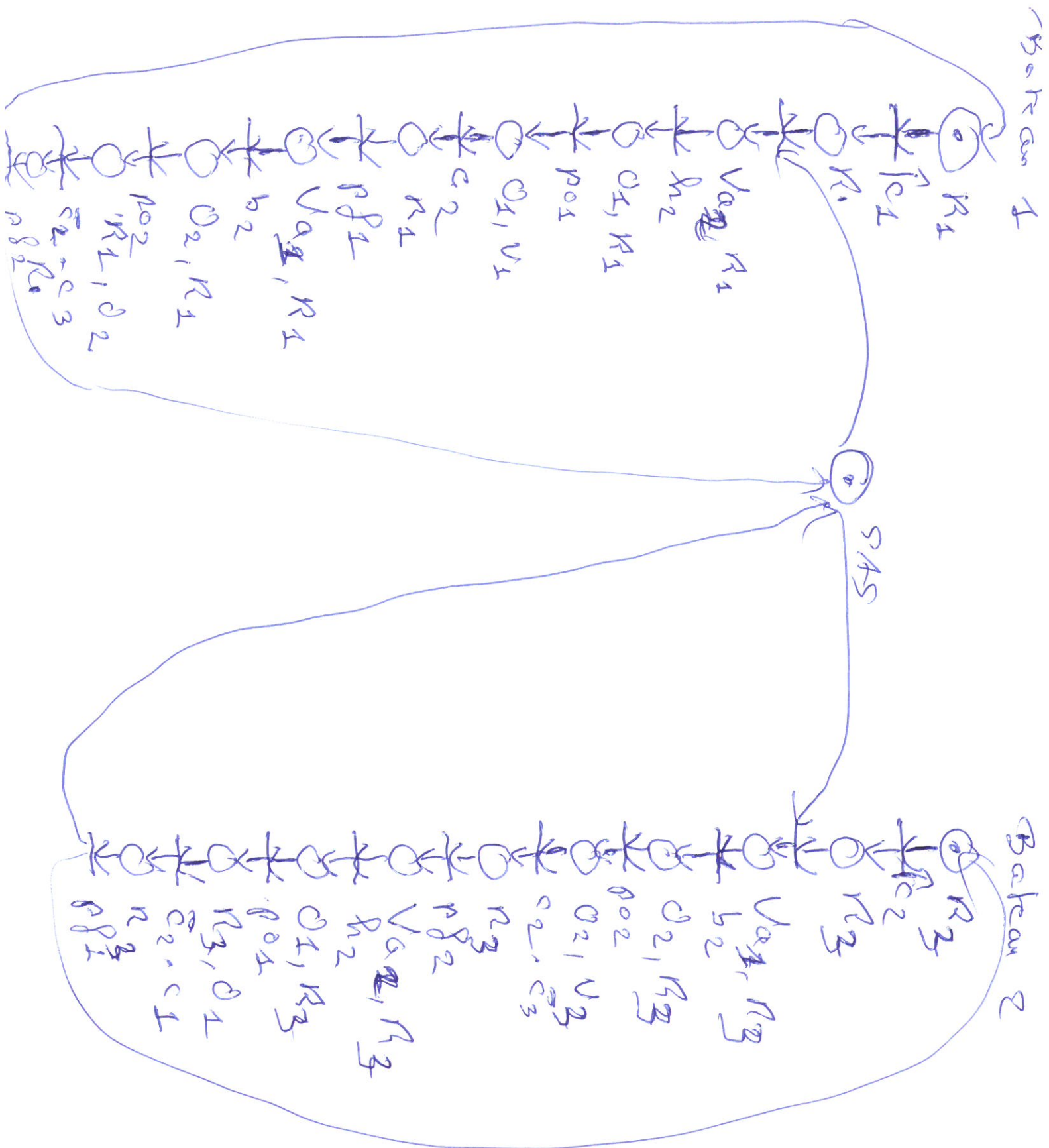


Figure 5. Notion de sémaphore en réseau de Petri ordinaire

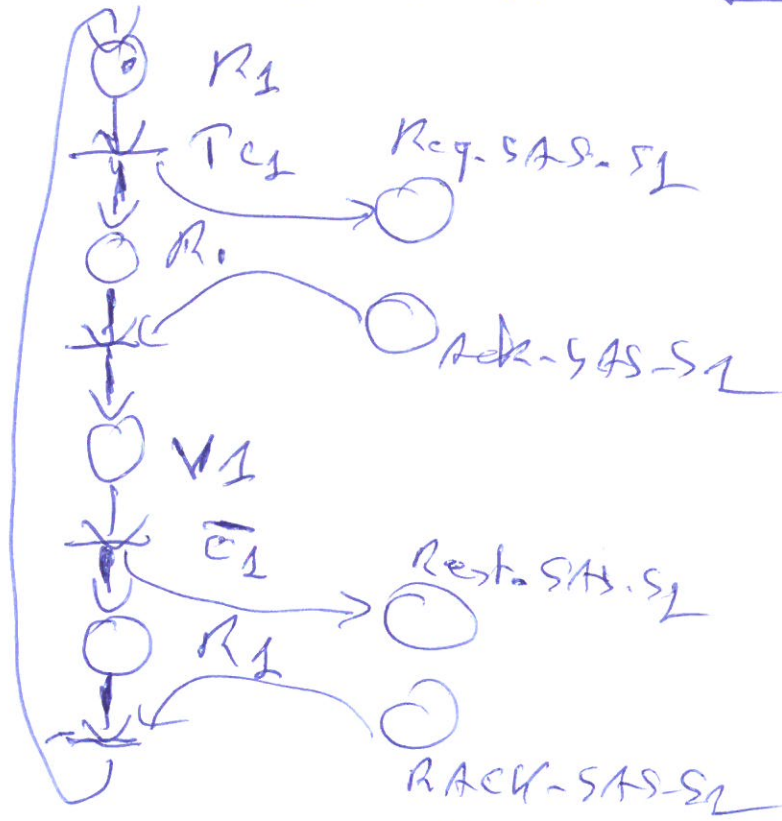
2. Modéliser en réseau de Petri ordinaire le processus de gestion de la ressource R1 (Allocateur de R1). On supposera que pour chaque tâche demandeuse, on utilise une paire de sémaphore pour l'envoi de la requête et la réception de l'affectation, et une autre paire pour la restitution.
3. Choisissez sur l'API3 des variables pour implémenter l'allocateur de la ressource R1.
4. Coder en PL7-3 l'allocateur de R1. On précisera notamment le code pour communiquer avec les API1 (adresse XWAY 1.1) et API2 (adresse XWAY 1.2).
5. Modéliser en réseau de Petri ordinaire le modèle de conception détaillé de commande du système compte-tenu de la distribution (On ne montrera pas la communication mais seulement le principe de la synchronisation par sémaphore).
6. Le système est-il vivant ? Justifier votre réponse.
7. Proposer une solution pour que le système soit parfaitement opérationnel.



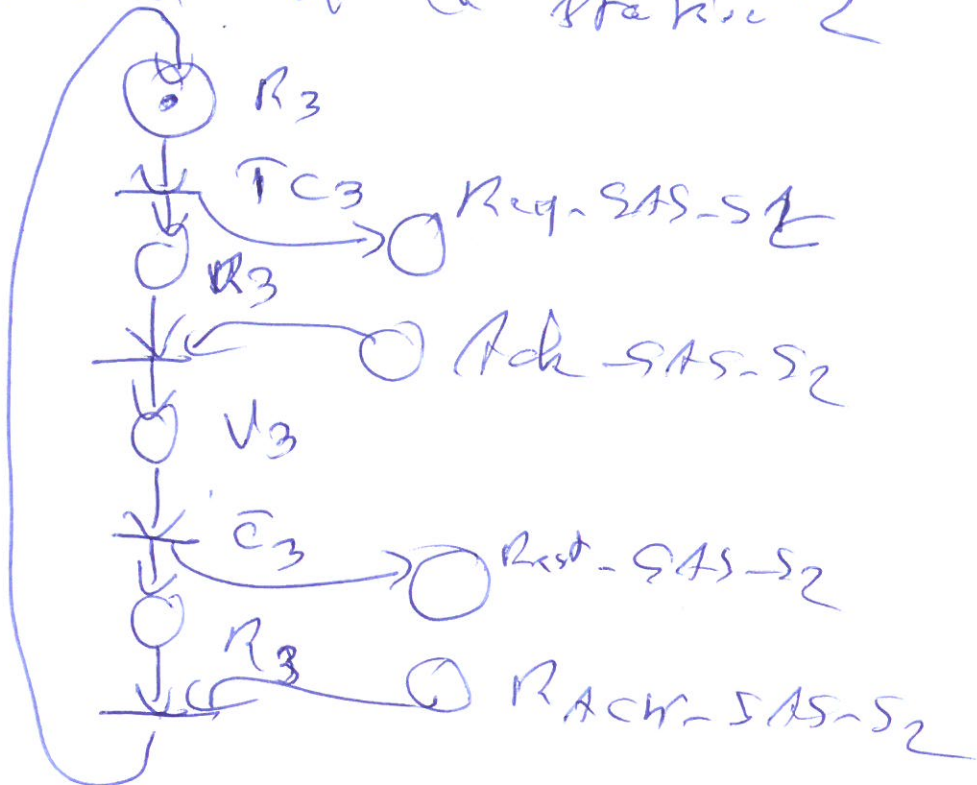


Exo 1 (suite 12)

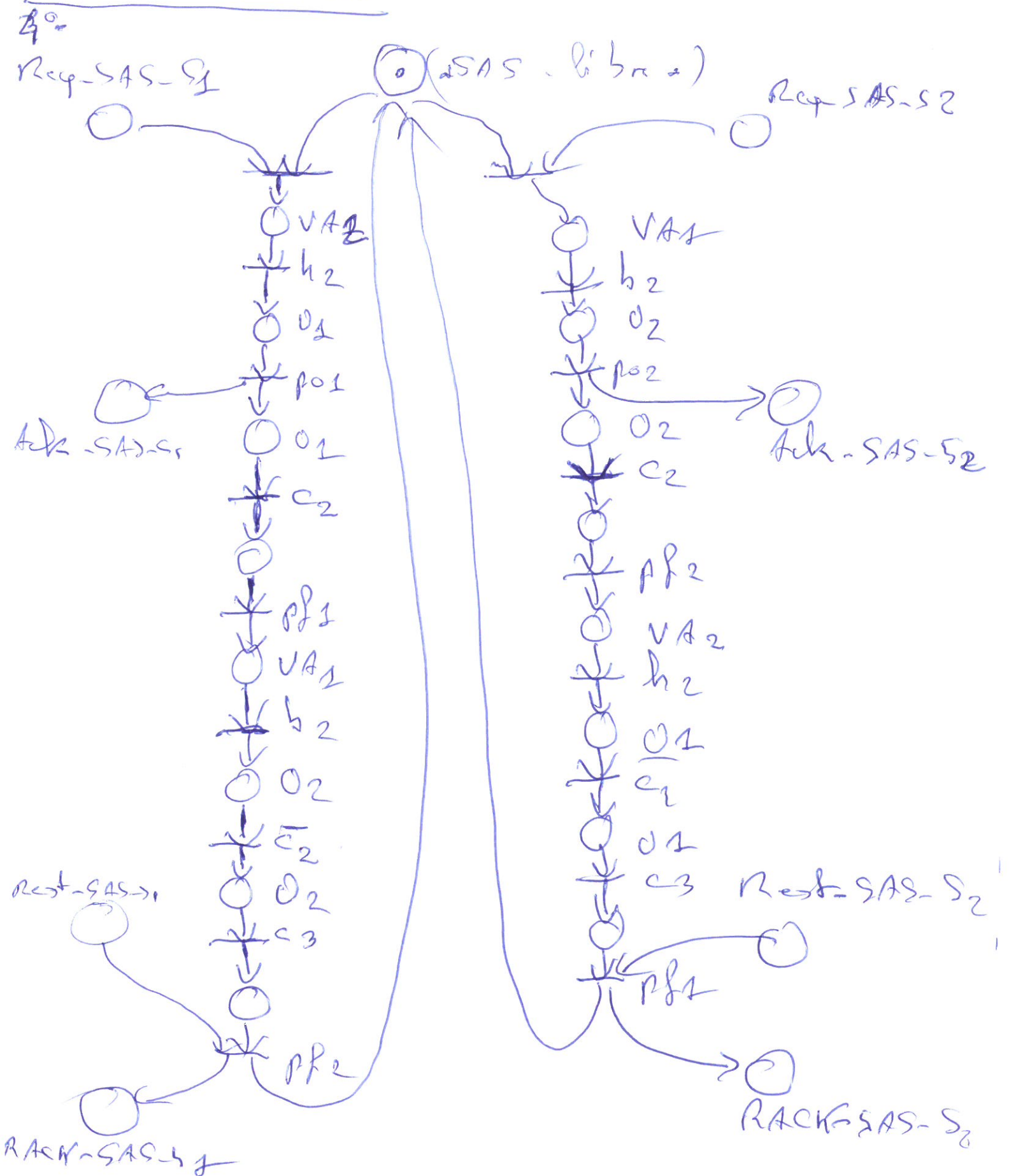
2° - Commande de la station 1



3° - Commande de la station 2

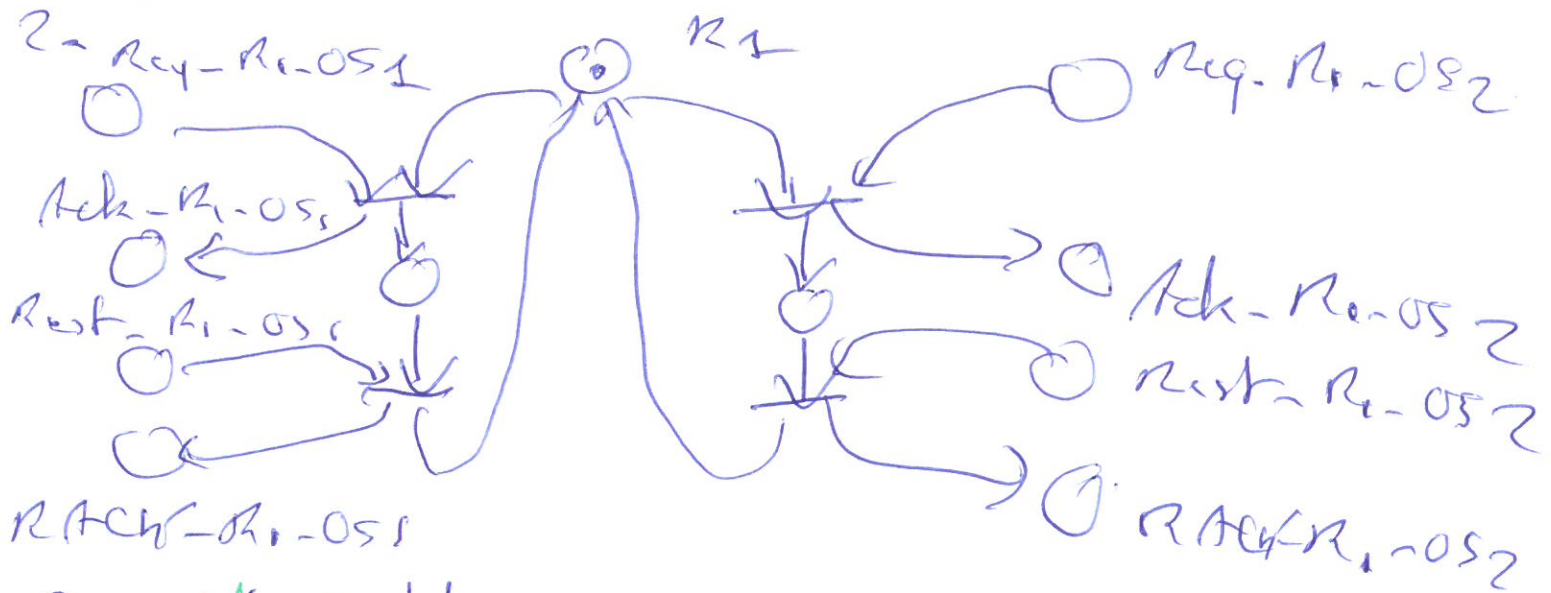


# Ex01 (saikuz'2)



## Exercice 2

1. L'asynchronisme de l'architecture opérationnelle de commande.

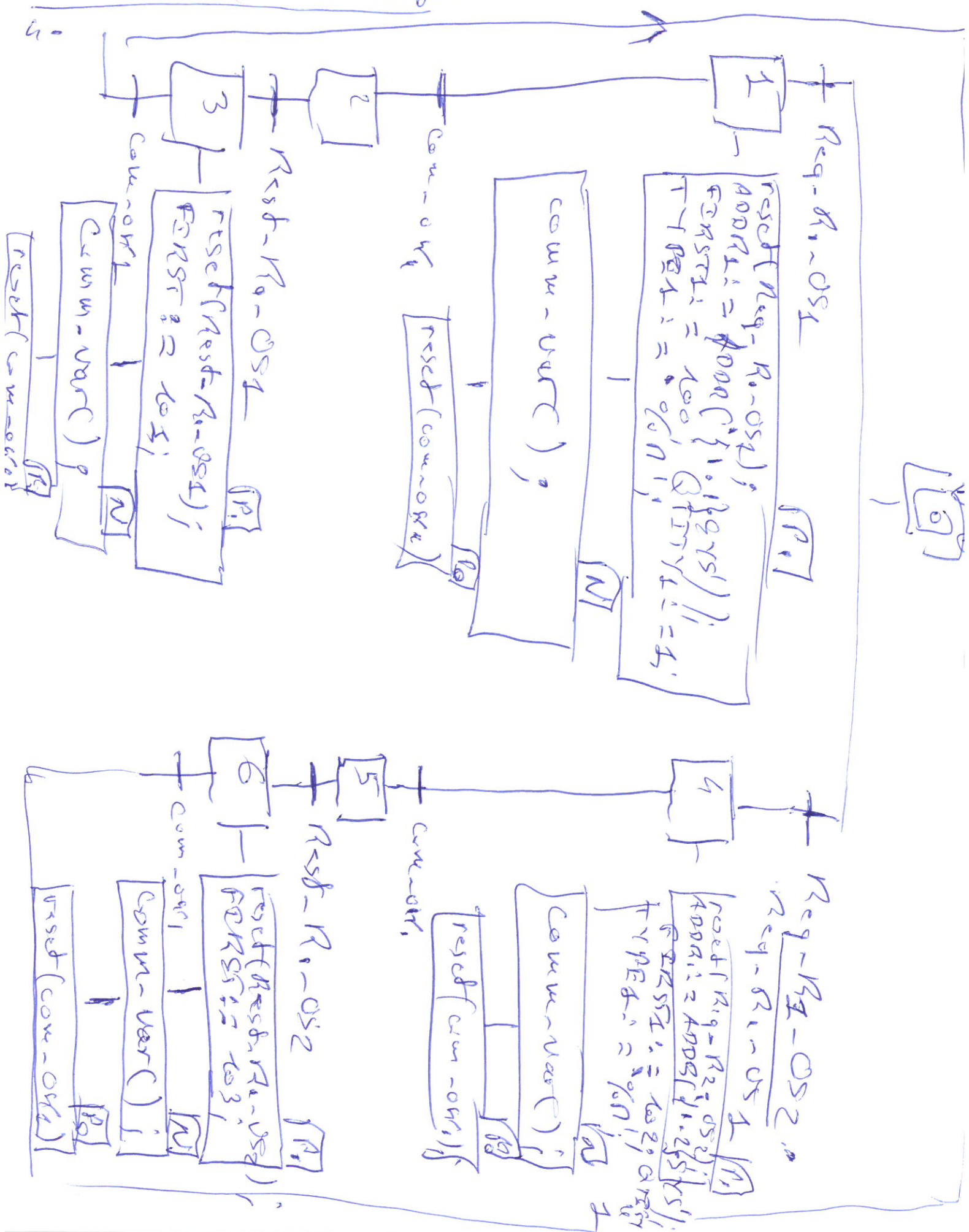


3. Variables sur AP23

Non Variables	Repeindre	Type
Req-R1-OS1	%P100	BOOL
Rest-R1-OS1	%P101	BOOL
Req-R1-OS2	%P102	BOOL
Rest-R1-OS2	%P103	BOOL
WMAXRETRY	%KW0	INT
ADDR1	%MW0-%MW5	INT
RETRY1	%RW6	INT
GRST1	%RW0-%RW3	INT
CR1	%RW11	INT
Act1	%RW0.0	BOOL
PLA1	%P10	BOOL



Exercice 2 (suite n° 2)



Exercice 2 (suite n°2)

40- (suite)

comm-var1) (\* Emission request \*)

if not act1 and not flag1 then

NOVE-ARRAY-INT(0, GEST1);

write-var(ADDR1, TYPE1, FIRST1, QTIY1, ENISS1, GEST1);

set(flag1); return; end-if;

(\* Test CR negative \*)

if not act1 and flag1 and retry1 < MAX\_RETRY

and CR1 < 20 then retry1 := retry1 + 1; reset(flag1);

return; end-if;

(\* Test CR positive \*) and CR1 = 20

if not act1 and flag1 then

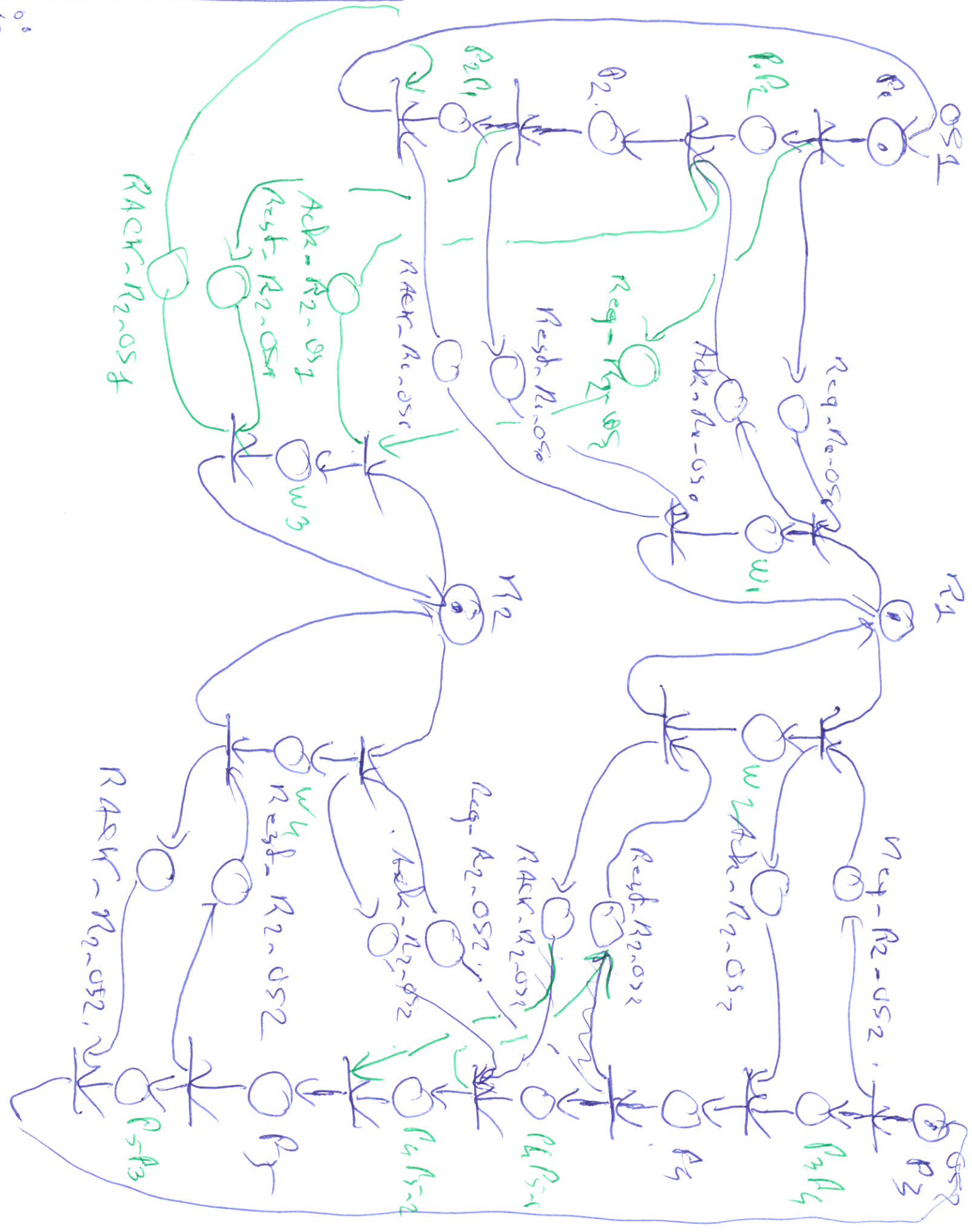
reset(flag1); set(comm-on1); return; end-if;

30- Variables (Suite)

Non variable	Représ	Type
comm-on1	%R11	EBool
FIRST1	%R10	DINT
QTIY1	%R12	DINT



50





# ⑧ TD2 LE3 Examen d'automatisme 2019-2020

## Exercice 2 (suite n°6)

Soit On: le système sera bloqué si  
ou à la configuration suivante:

$$\Pi(w_2) = 1; \Pi(P_4 P_5 = 1); \Pi(Req. R_2 - OS_2) = 1;$$

$$\Pi(w_3) = 1; \Pi(Req. R_4 - OS_4) = 1; \Pi(P_1 P_2) = 1;$$

$$\Pi(Ack. R_2 - OS_2) = 1;$$

7a Pour éviter le blocage, il faut  
prendre les 2 ressources en même  
temps.

