

République Tunisienne
Ministère de l'Education

GÉNIE ELECTRIQUE

4^{ème} année de l'enseignement secondaire
Sciences Techniques

Manuel d'activités

Les auteurs

Ammar MZOUGH
*Inspecteur Principal
des lycées et collèges*

Ali ZITOUNI
*Inspecteur
des lycées et collèges*

Hatem LABIDI
Technologue

Mounir BEN HENDA
*Professeur Principal
hors classe*

Fethi AYARI
Professeur Principal

Les évaluateurs

Fredj JAZI
*Inspecteur Général
de l'éducation*

Mohamed BEN HMIDA
*Inspecteur Général
de l'éducation*

Fateh BJAOU
Technologue

Avant-propos

Si le vingtième siècle a connu la révolution industrielle, le vingt-et-unième s'annonce comme celui où l'électronique digitale associée à la logique programmée vont transformer tous les secteurs de l'activité humaine, qu'il s'agisse des secteurs de télécommunication, de l'audio-visuel, de l'électroménager, de l'automobile ou de l'aviation.

Par comparaison avec l'ancien programme, toutes les innovations dans la discipline se rapportent à ce secteur technologique d'actualité, dans ce cadre on cite en particulier :

- ✓ *L'arithmétique binaire : addition, soustraction, multiplication et division.*
 - ✓ *Les circuits combinatoires : la mise en œuvre d'une unité arithmétique et logique.*
 - ✓ *La logique programmée : programmation des microcontrôleurs de Microchip par la mise en œuvre d'un langage évolué dont la structure générale est familière à l'élève.*
- Une bonne partie de ce manuel est consacrée à ce type de thèmes.*

Cet ouvrage est conçu dans le but de servir en tant que document ressources pour le professeur et guide personnel pour l'élève qui lui permet de découvrir ou de vérifier certains concepts ou méthodes de résolution.

Tout au long de ce manuel, nous avons cherché la simplicité et la faisabilité.

✓ *Simplicité : aucune connaissance préalable n'est requise ; les prérequis scolaires de l'élève de la 1^{ère} année secondaire à la 3^{ème} technique suffisent largement.*

✓ *Faisabilité : l'aspect pratique est partout présent, il est le pilier principal de l'appropriation du savoir enseigné. Les activités proposées sont réalisables en autonomie relative par l'élève lui-même. Le rôle de l'enseignant se limite à l'accompagnement.*

Les résultats des travaux de l'élève ont pour but de lui faciliter la découverte, l'appropriation et la validation des concepts nouveaux.

Les auteurs de cet ouvrage restent à l'écoute de toutes les remarques et les suggestions des utilisateurs de ce manuel, dans le but d'apporter les améliorations et les corrections requises dans les éditions futures.

Les Auteurs

Sommaire

	Pages
Avant-propos	3
Sommaire	4
<i>PARTIE AUTOMATIQUE</i>	
Chapitre A1 : Opérations arithmétiques binaires	
TP A1-1 : Addition et Multiplication en binaire	6
TP A1-2 : Soustraction et Division en binaire	28
Chapitre A2 : Logique combinatoire	
TP A2-1 : Les comparateurs logiques	36
TPA2-2 : Unité arithmétique et logique	43
Chapitre A3 - Logique séquentielle	
TP A3-1 : Les compteurs	48
Chapitre A4 - Logique programmée	
TP A4-1 : Automate Programmable Industriel (API)	56
TP A4-2 : Microcontrôleurs	74
Chapitre A5 Notions d'asservissement linéaire	
TP A5-1 : Asservissement linéaire	99
<i>PARTIE ELECTROTECHNIQUE</i>	
Chapitre B1 : Les systèmes triphasés	
TP B1 : Les systèmes triphasés	106
Chapitre B2 : Les moteurs asynchrones triphasés	
TP B2 : Les moteurs asynchrones triphasés	112
Chapitre B3 : Les moteurs à courant continu	
TP B3 : Les moteurs à courant continu	117
<i>PARTIE ELECTRONIQUE</i>	
Chapitre C1 : Amplificateurs linéaires intégrés	
TP C1 : Les amplificateurs linéaires intégrés	124

TP A1-1

L'Addition et la Multiplication en Binaire

- Objectifs Spécifiques :

OS A1-1 - Exécuter en binaire une opération arithmétique de base

OS A1-2 - Représenter un nombre entier relatif

OS A2-2 - Mettre en œuvre un circuit arithmétique.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Poste PC
- ✓ Logiciel de simulation
- ✓ Matériel d'essai en électronique numérique
- ✓ Simulateur logique
- ✓ Maquette.

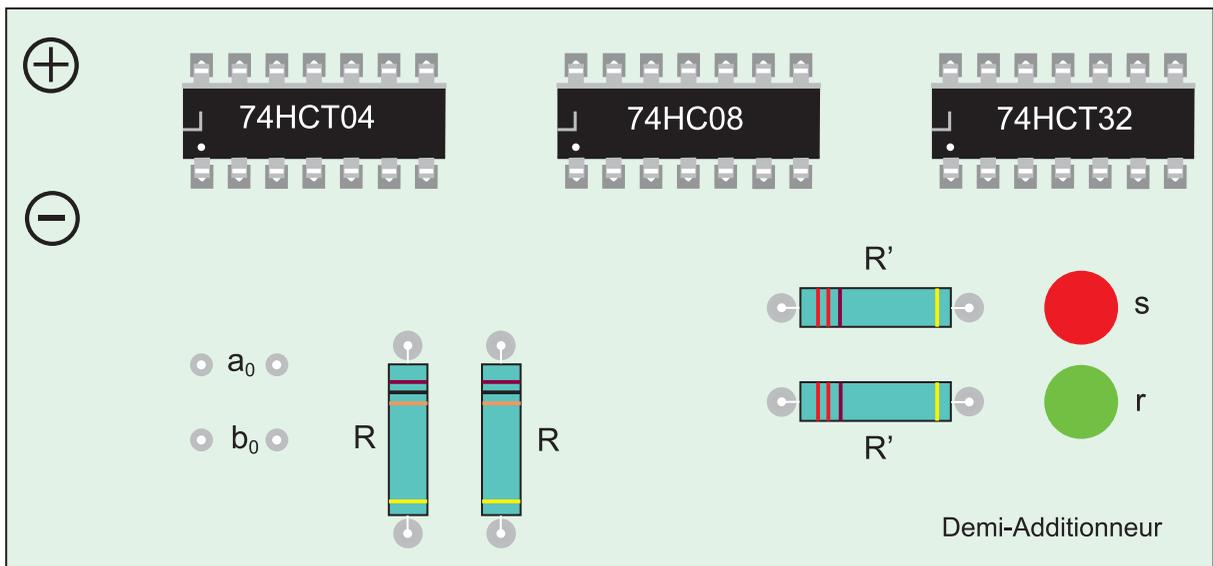


Activité 1

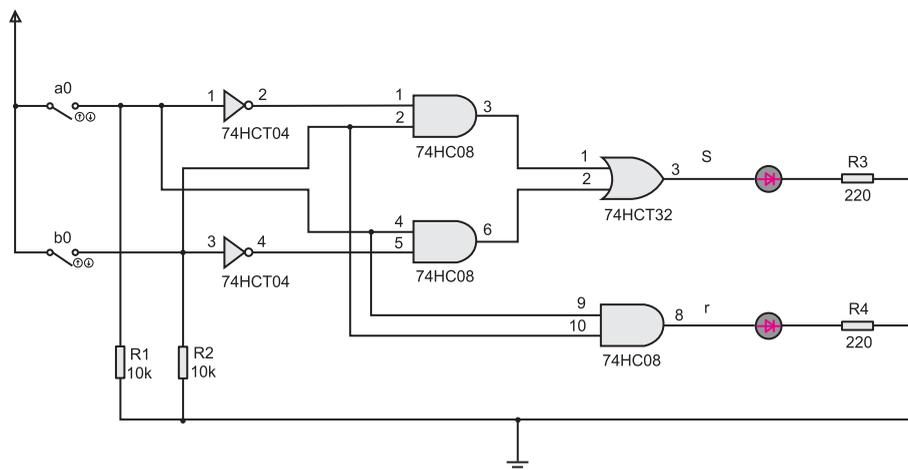
Cette activité consiste à chercher la somme de deux nombres binaires à un bit chacun ($A : a_0$) et ($B : b_0$), la retenue qui peut en découler suite à cette opération et par la suite dégager la ou les limites de ce circuit.

1- En utilisant le Datasheet du constructeur "fourni par le professeur ou récupéré sur un site Internet", identifier les circuits figurant sur la platine ci-dessous.

- ✓ 74HCT04 :
- ✓ 74HC08 :
- ✓ 74HCT32 :



2- Compléter le câblage de la platine ci-dessus, conformément au schéma ci-dessous.





6- Proposer une solution à base de portes logiques universelles **NAND** ou **NOR** (question facultative).



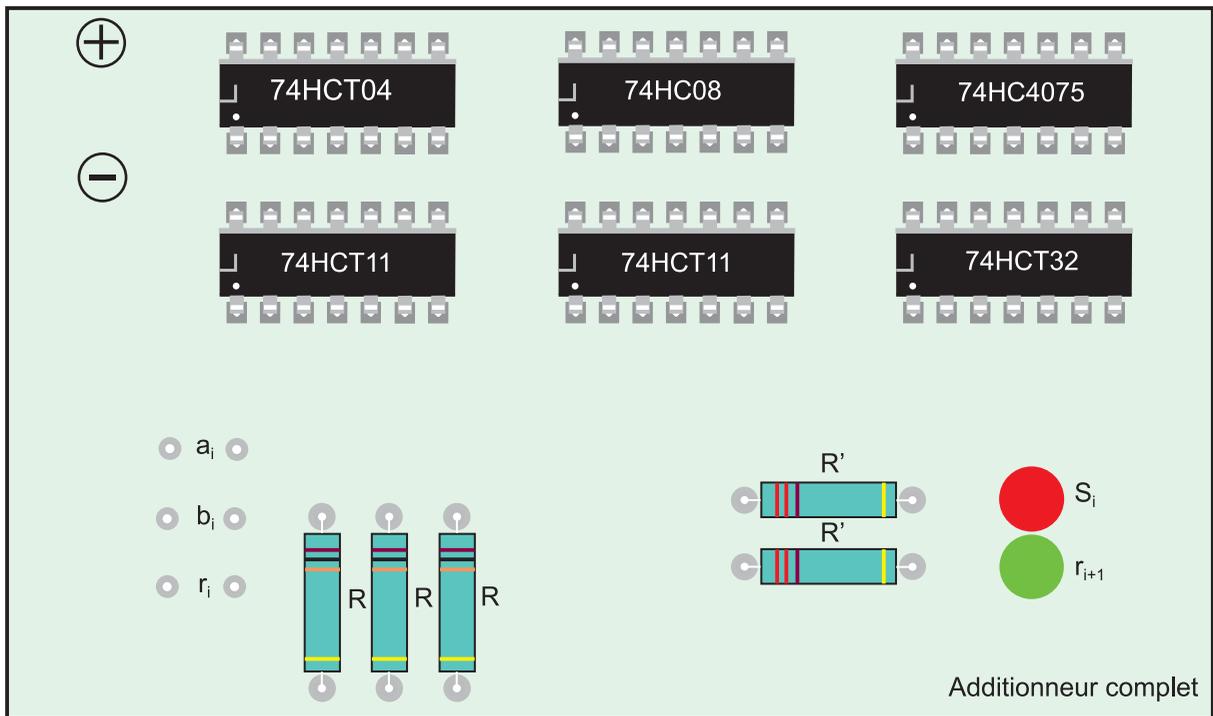
Activité 2

Cette activité consiste à remédier au défaut rencontré lors de l'étude du **demi-additionneur**.

1- En utilisant le Datasheet du constructeur "fourni par le professeur ou récupéré sur un site Internet", identifier les circuits figurant sur la platine ci-dessous.

✓ 74HCT11 :

✓ 74HC4075 :



2- Compléter le câblage de la platine ci-dessus, conformément au schéma présenté ci-après.

Légende :

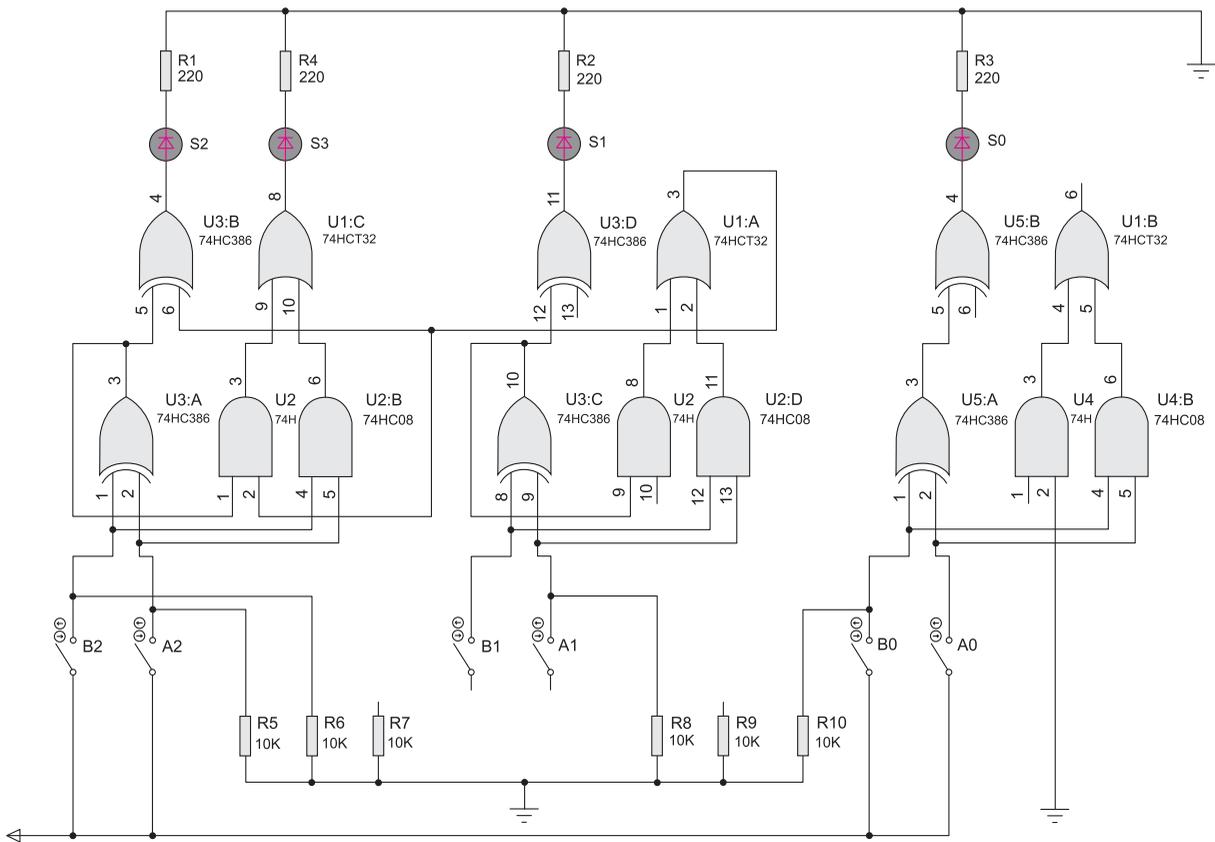
- ✓ a_i , b_i et r_i : entrées.
- ✓ S_i : Led rouge signalant l'état de la somme.
- ✓ r_{i+1} : Led verte signalant l'état de la retenue.
- ✓ R : résistance de forçage ou de rappel (4,7 à 10K ohm)
- ✓ R' : résistance de limitation.
- ✓ **74XX** : CI.



Activité 3

L'activité consiste à réaliser un additionneur de deux nombres à 3 bits (**A** : $a_2a_1a_0$), (**B** : $b_2b_1b_0$) en utilisant des additionneurs élémentaires complets (à base de cellules logiques de base) de deux nombres à 1 bit chacun branchés en cascade.

- 1- Compléter le logigramme suivant pour en faire un additionneur binaire de deux nombres à 3 bits : (**A** : $a_2a_1a_0$ et **B** : $b_2b_1b_0$).





2- En utilisant le Datasheet du constructeur "fourni par le professeur ou récupéré sur un site Internet", identifier le circuit "74HC386" :

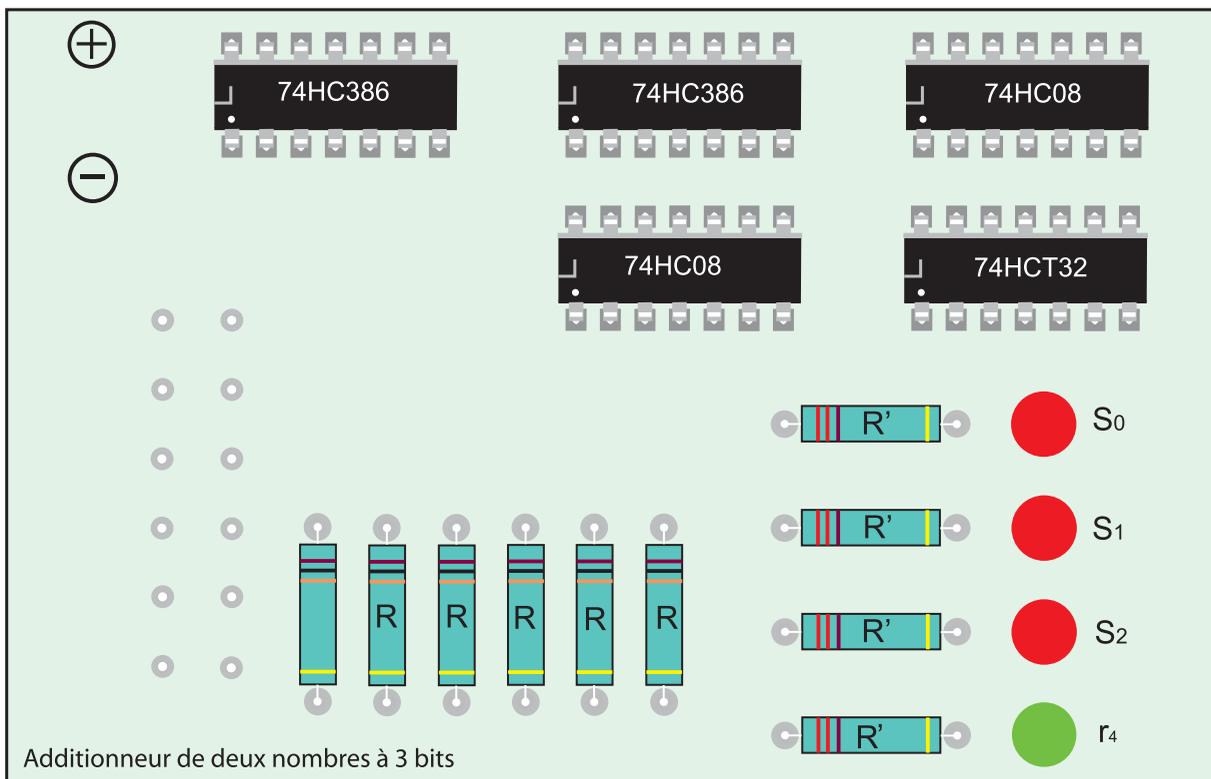
3- Compléter :

3.1- La légende ci-contre

Légende :

- ✓ $a_0, a_1, \text{ et } a_2$:
- ✓ $b_0, b_1, \text{ et } b_2$:
- ✓ $S_0, S_1 \text{ et } S_2$:
- ✓ r_4 :
- ✓ R :
- ✓ R' :
- ✓ 74HC08 :
- ✓ 74HCT32 :
- ✓ 74HC386 :

3.2- le cablage de la platine ci-dessous en utilisant le schéma proposé à la question "N° 1".



4- En utilisant une maquette, le simulateur logique ou le logiciel de simulation, vérifier le fonctionnement de ce montage.



Activité 4

Conception d'un additionneur de deux nombres à deux bits ($A : a_1a_0$ et $B : b_1b_0$) à retenue anticipée.

1 - Circuit d'anticipation :

1.1- Calculer les différentes retenues, sachant que la retenue à l'entrée du premier additionneur élémentaire est égale à zéro ($r_0 = 0$).

A large grid of 20 columns and 15 rows, intended for calculating carry bits for a 2-bit adder.

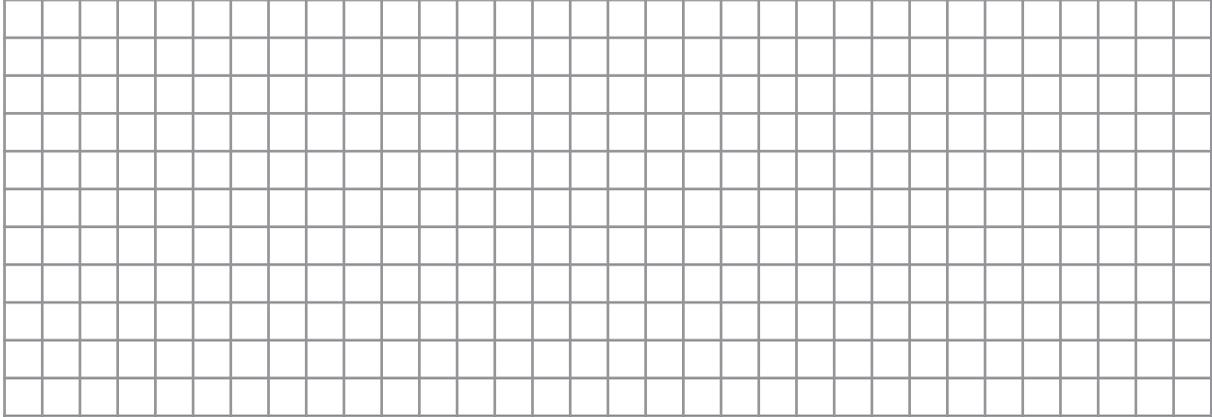
1.2- Déduire le logigramme correspondant.

A large grid of 20 columns and 15 rows, intended for drawing a logic diagram for the carry-look-ahead adder.

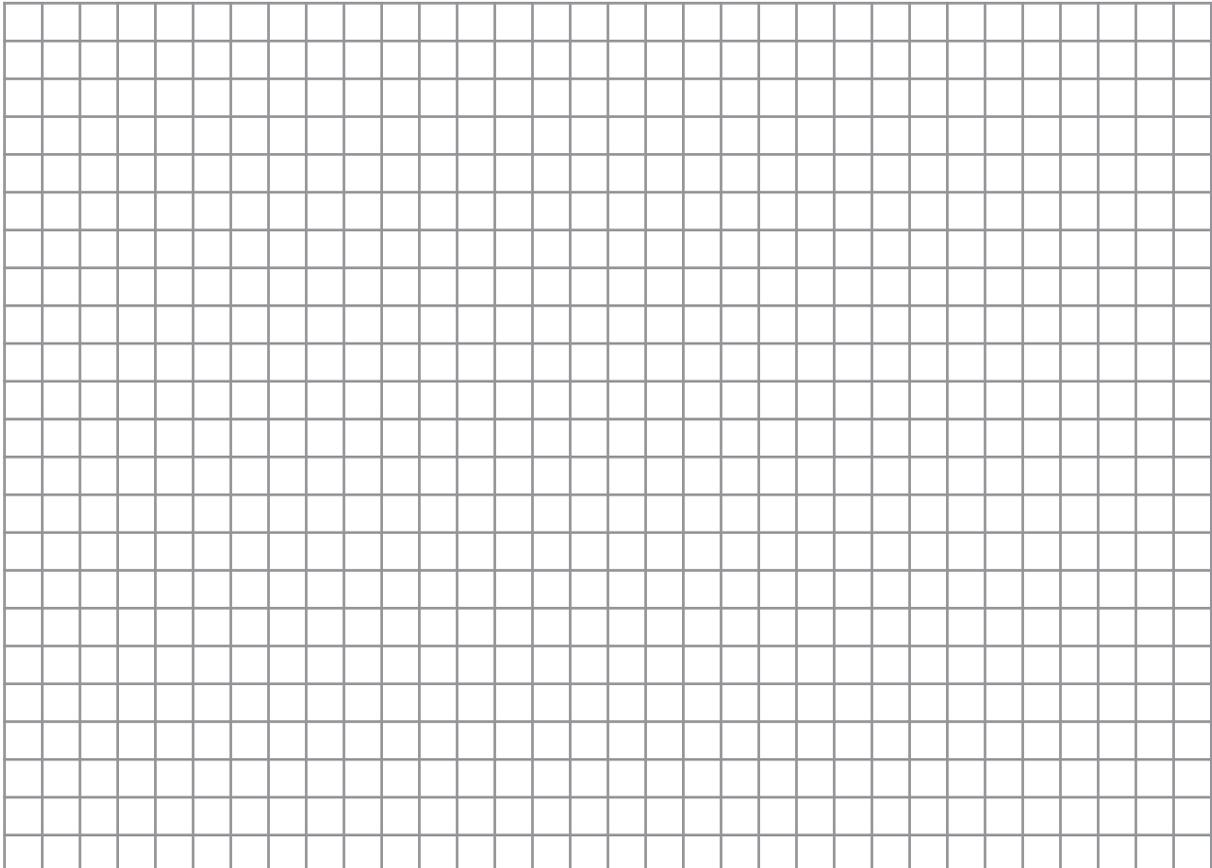


2 - L'additionneur complet :

2.1- Dédire les différentes équations de l'additionneur.



2.2- Dédire le logigramme correspondant.

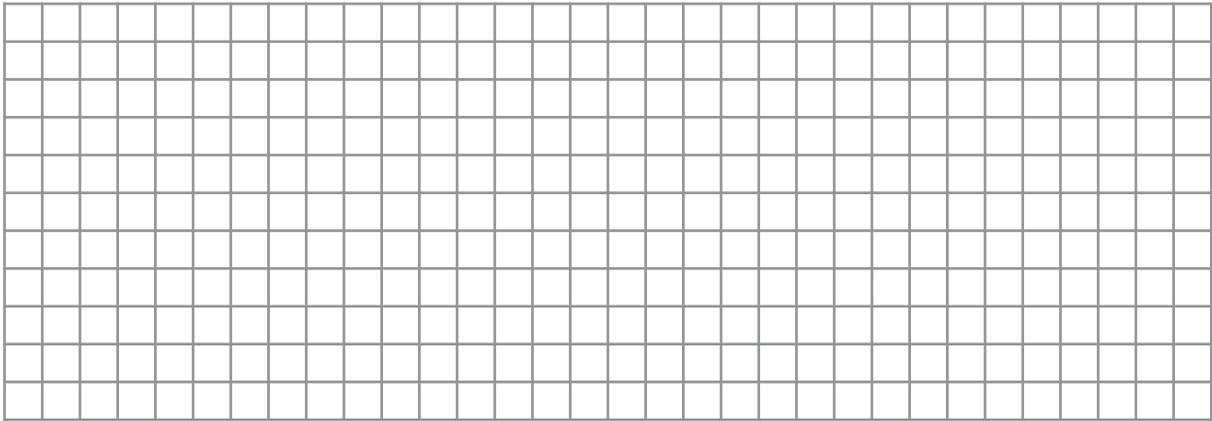


2.3- Sur logiciel de simulation, câbler le logigramme tracé précédemment et vérifier son fonctionnement.



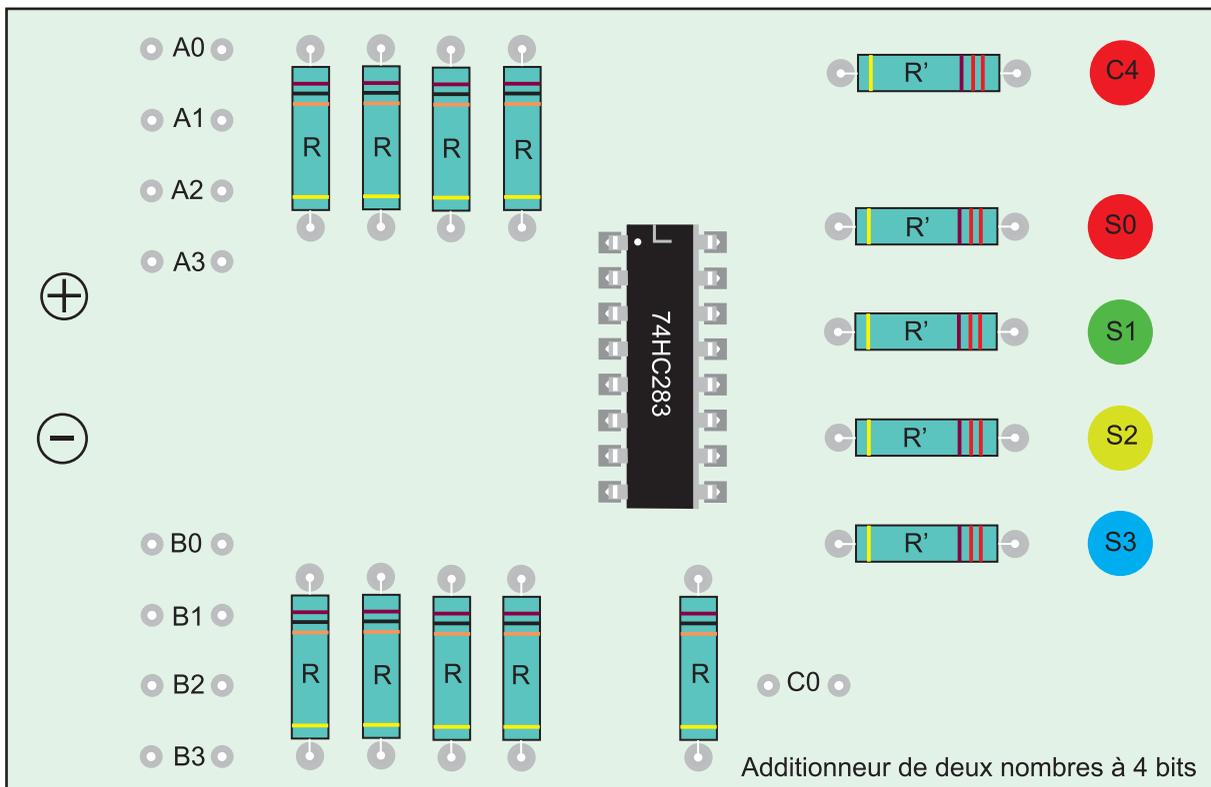
3- Mise en œuvre d'un additionneur intégré (choisir un circuit de la série **74HCxx**).

3.1- En utilisant le Datasheet relatif au circuit objet de l'étude, déduire ses principales caractéristiques.



3.2- Sur maquette, simulateur ou avec un logiciel de simulation :

a. Câbler la platine ci-dessous conformément au schéma structurel joint :

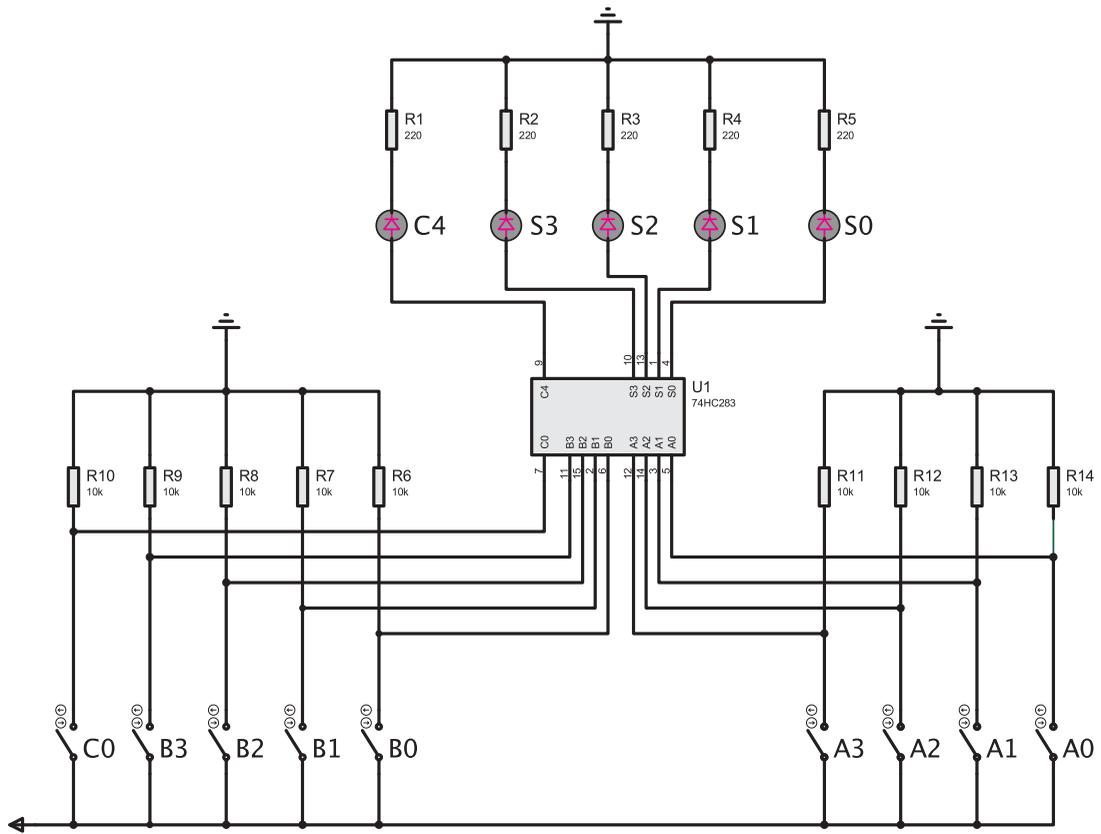


Légende :

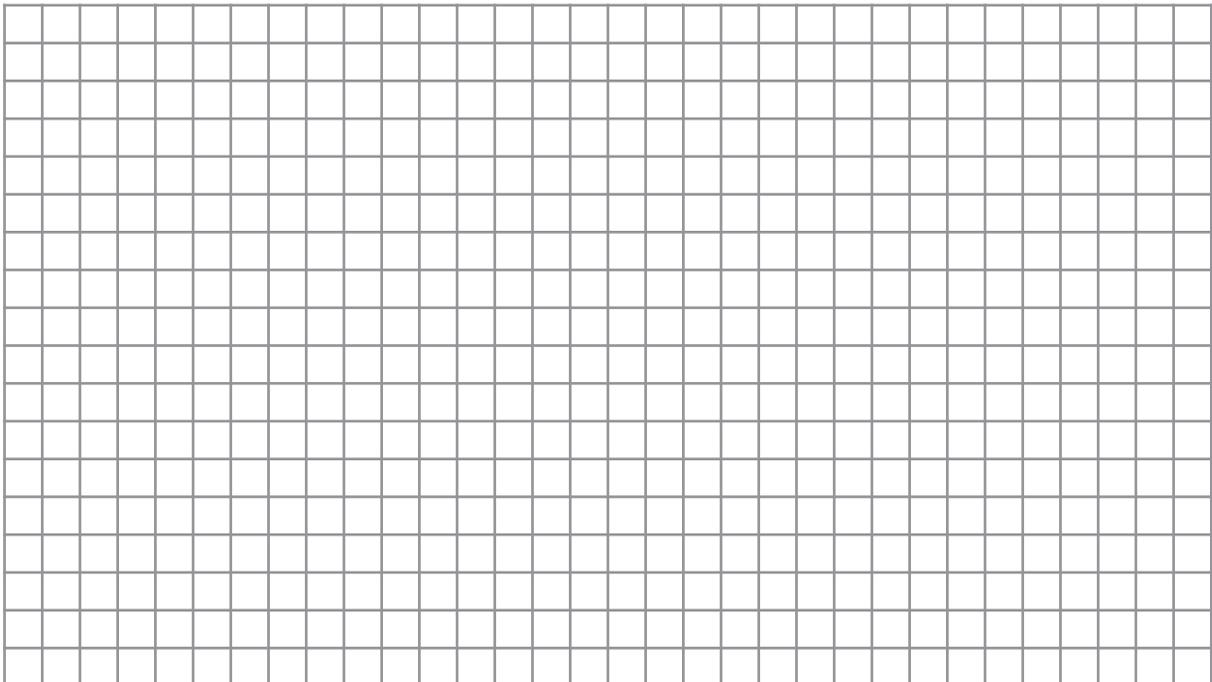
- ✓ Ai et Bi :
-
- ✓ 74HC283 :

Légende :

- ✓ R et R' :
-
- ✓ C0 :



- b. Vérifier son fonctionnement
- c. Comment faut-il câbler ce circuit pour le faire fonctionner en additionneur de deux nombres à deux bits seulement ?

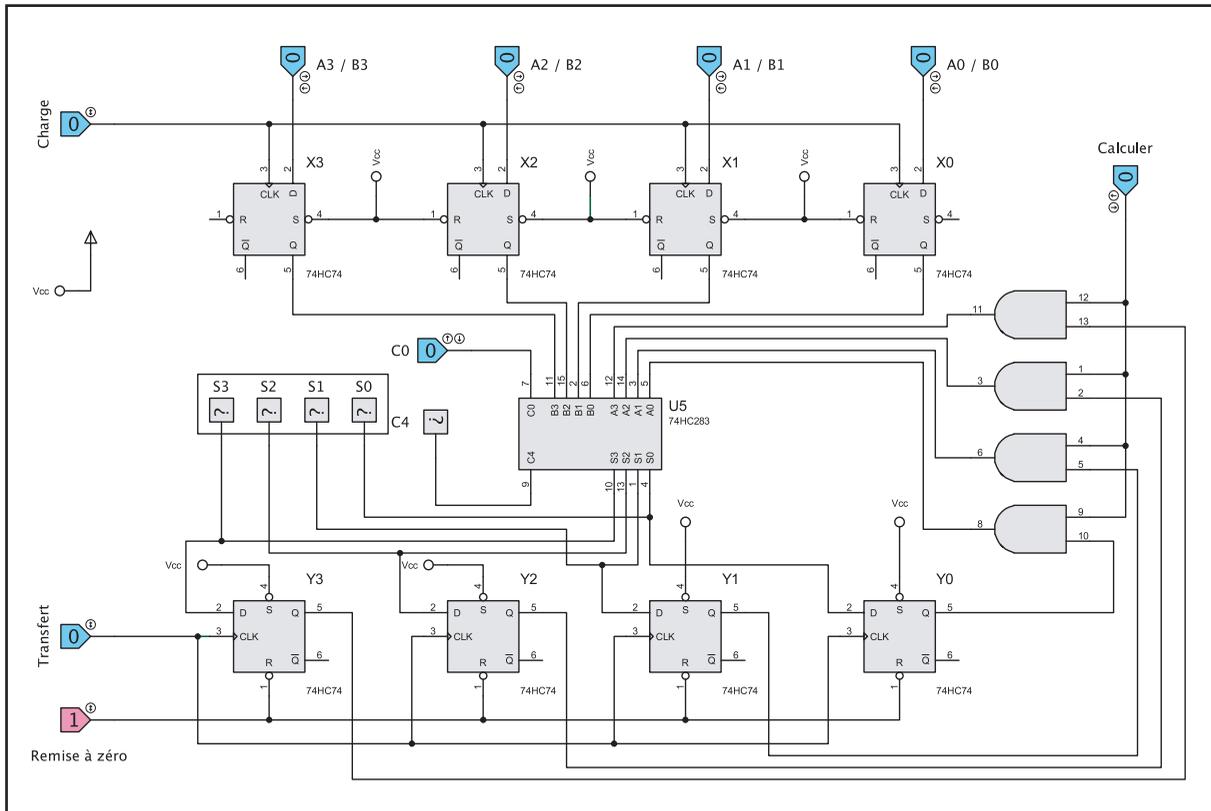




Activité 5

découverte des mécanismes mis en œuvre par un additionneur au cours de la réalisation d'une opération d'addition de deux nombres binaires.

Le schéma structurel suivant représente un additionneur binaire avec ses périphériques.



Légende :

- ✓ 74HC74 :
- ✓ 74HC283 :

Légende :

- ✓ Ai / Bi :
- ✓ Si :

1- En utilisant les documents constructeurs relatifs aux composants constituant ce schéma (documents récupérés sur sites Internet ou fournis par le professeur), remplir le tableau ci-dessous.

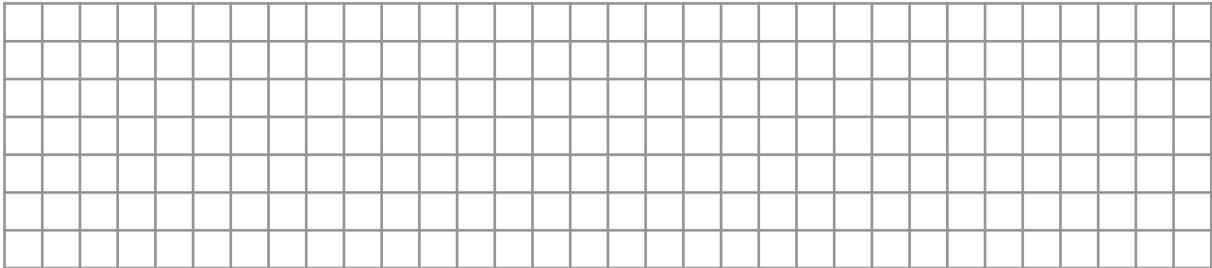
Circuit	Référence	Technologie	Fonction	Nom
Xi				
U5				



Activité 6

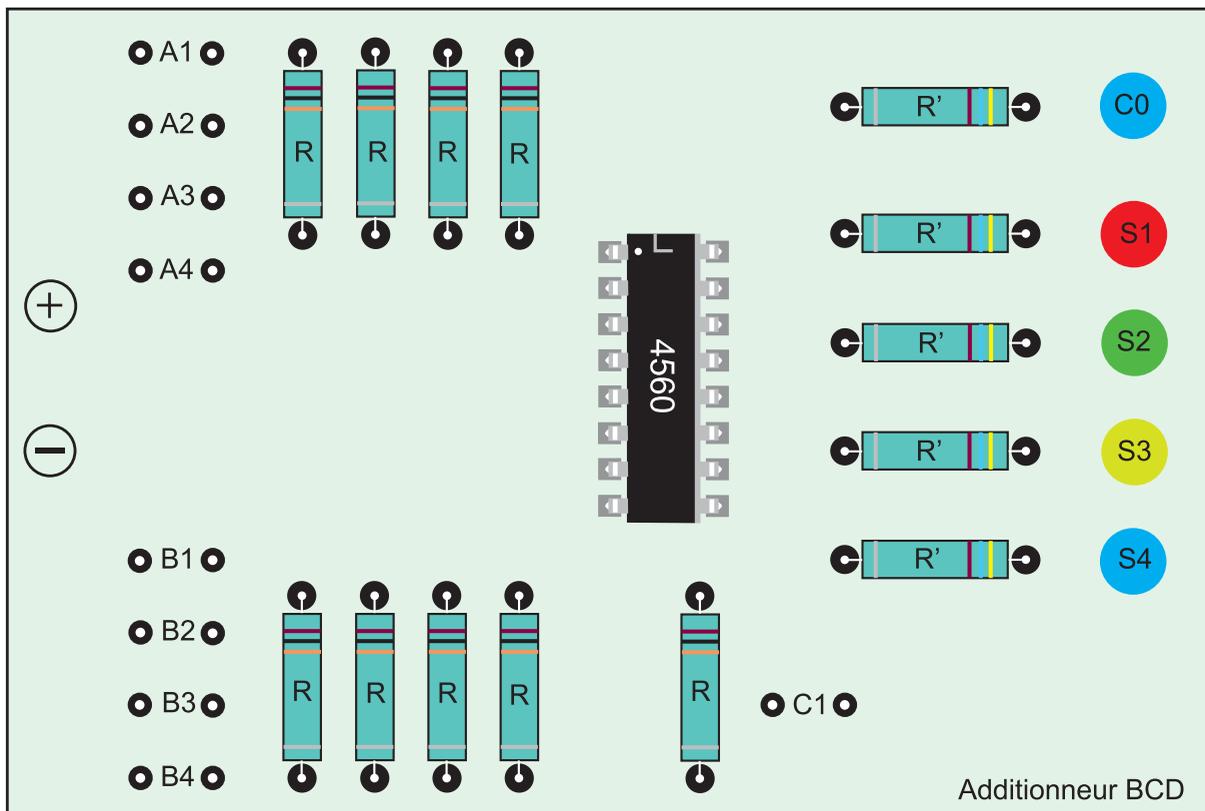
Mise en œuvre d'un circuit de la série 4560 " additionneur BCD "

1- A partir du datasheet déduire les principales caractéristiques du circuit.



2- Sur maquette, simulateur ou avec un logiciel de simulation :

a. Câbler la platine ci-dessous conformément au schéma structurel joint :



Légende :

✓ Ai et Bi :

.....

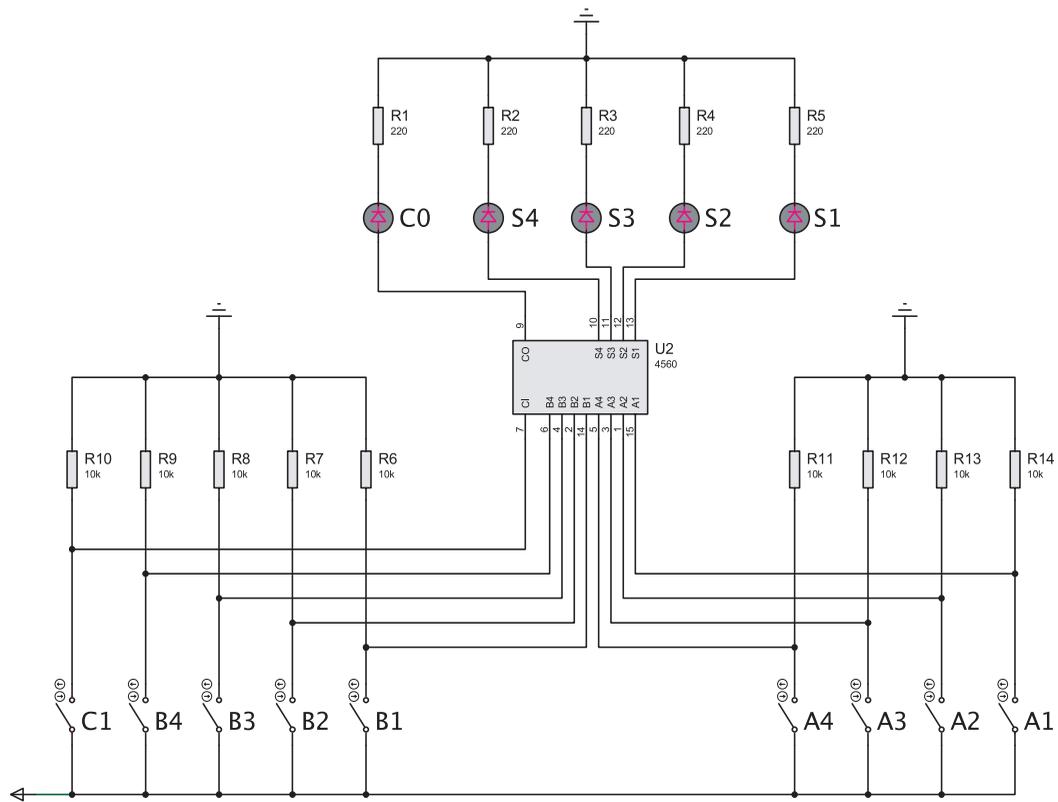
✓ 4560 :

Légende :

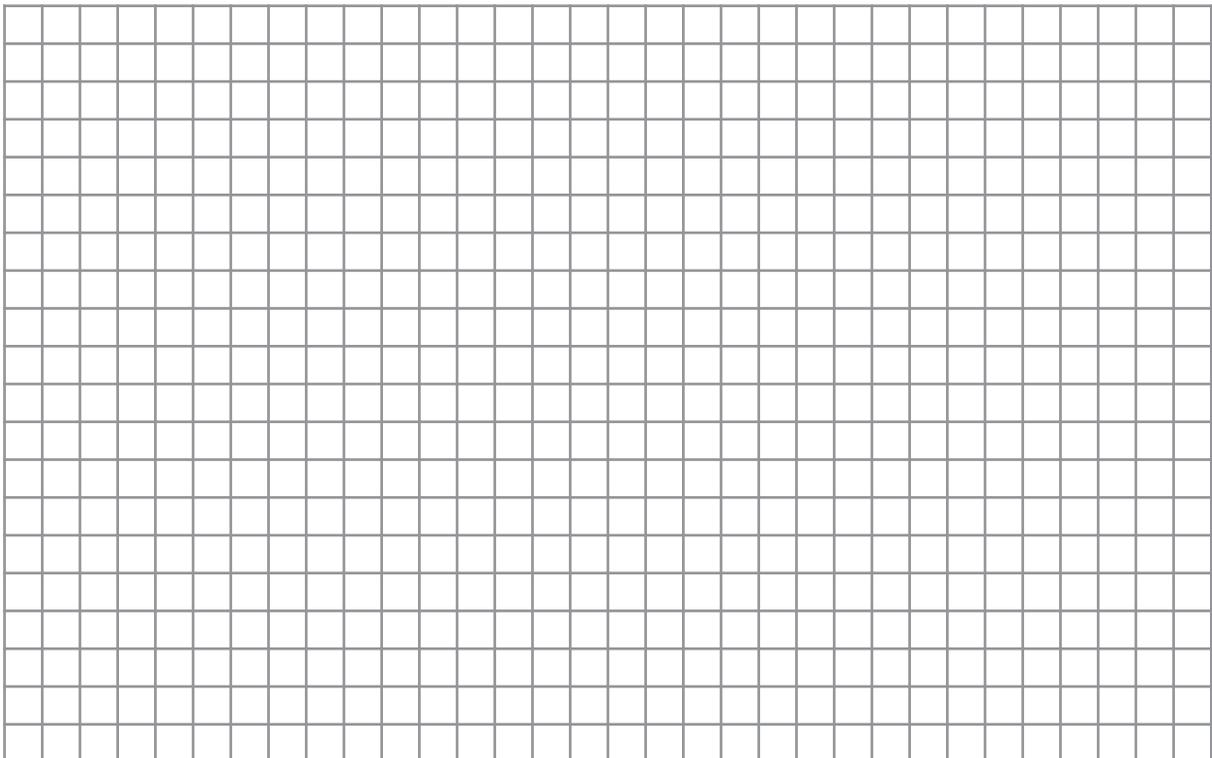
✓ R et R' :

.....

✓ Ci :

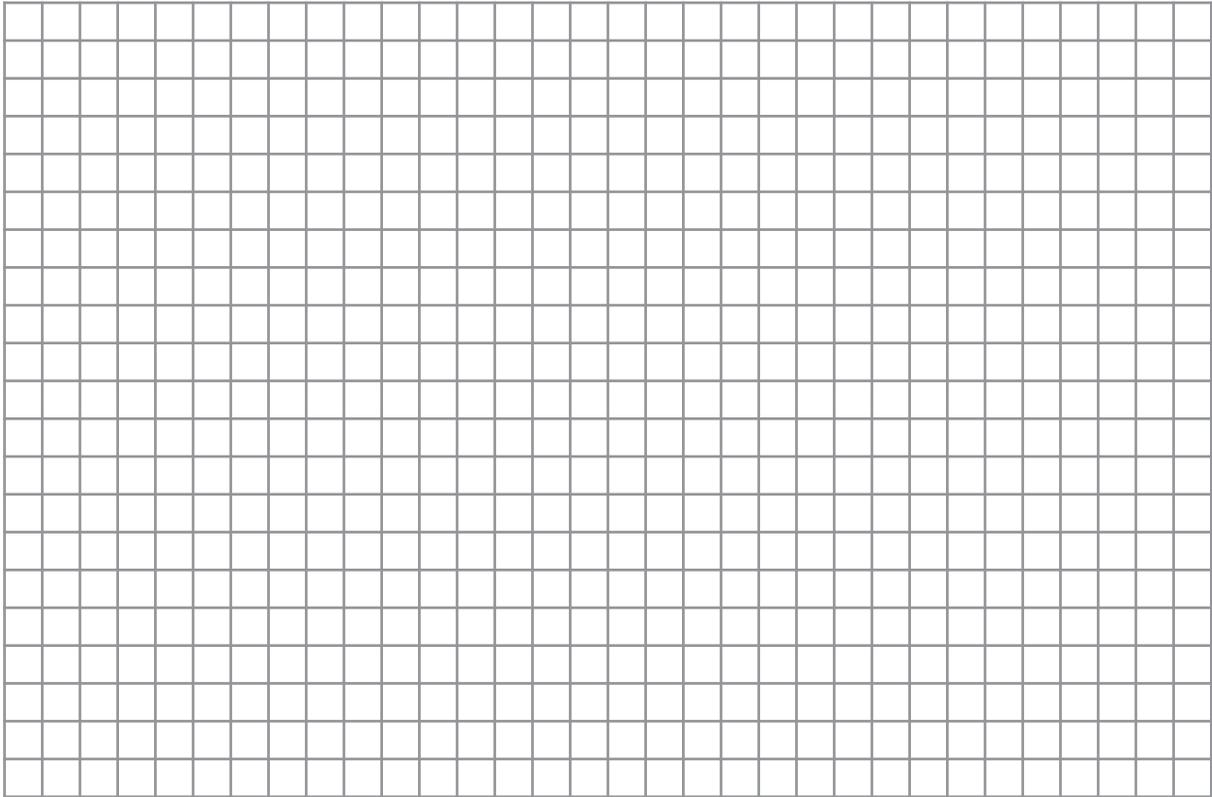


b. Vérifier le fonctionnement du circuit :



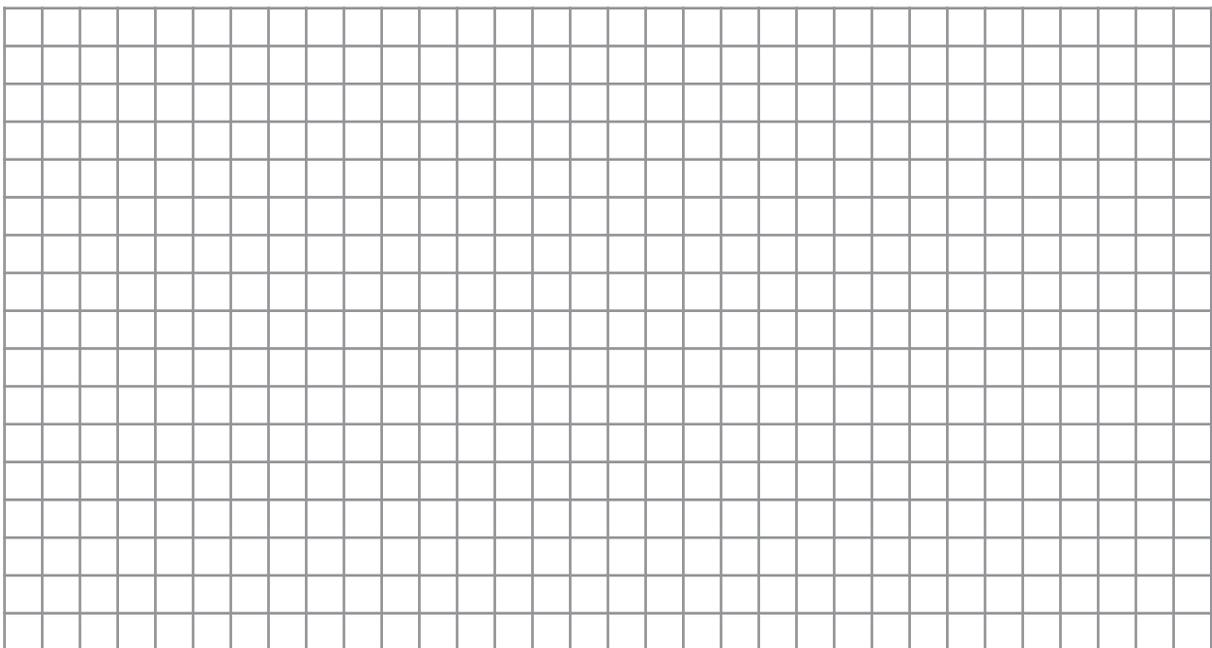


3- Tracer le logigramme correspondant :



4- Sur maquette, simulateur ou logiciel de simulation.

- a. Câbler la solution trouvée.
- b. Vérifier son fonctionnement.

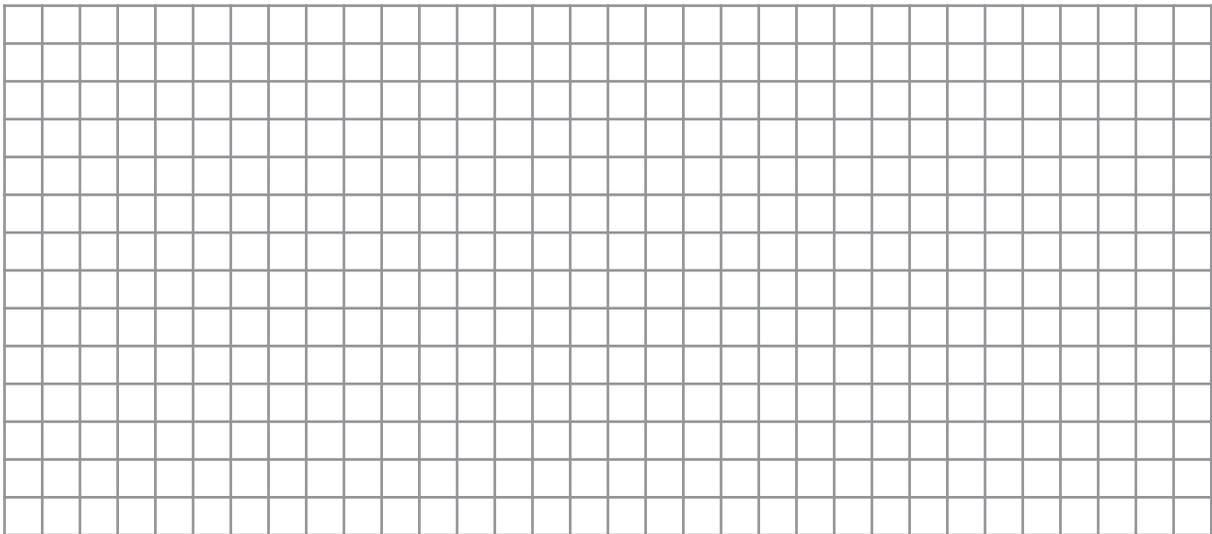




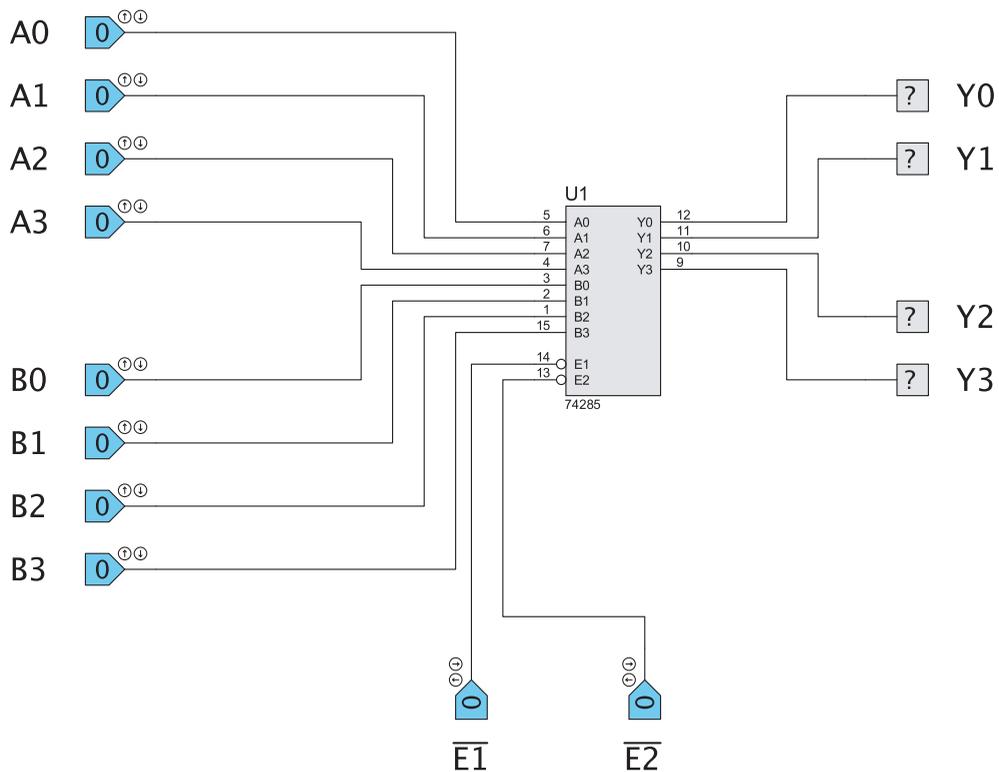
Activité 8

Il s'agit de simuler le fonctionnement un circuit multiplieur de deux nombres binaires à 4 bits de la serie 74xx soit le circuit intégré 74285.

- 1- En utilisant le Datasheet du constructeur "fourni par le professeur ou récupéré sur un site Internet", relever les principales caractéristiques de ce circuit.



- 2- Avec le logiciel de simulation, saisir le schéma suivant et verifier son fonctionnement :



TP A1-2

La Soustraction et la Division en Binaire

- Objectifs Spécifiques :

- OS A1-1 - Exécuter en binaire une opération arithmétique de base
- OS A1-2 - Représenter un nombre entier relatif
- OS A2-2 - Mettre en oeuvre un circuit arithmétique

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Poste PC
- ✓ Logiciel de simulation
- ✓ Matériel d'essai en électronique numérique
- ✓ Simulateur logique
- ✓ Maquette.

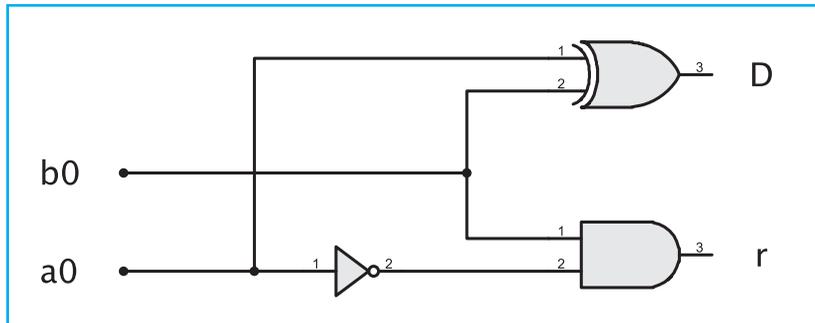


Activité 1

Il s'agit de cabler ou simuler le fonctionnement d'un circuit à base de cellules logiques, de tracer sa table de fonctionnement et de chercher les équations des sorties en vue de vérifier qu'il s'agit d'un demi-soustracteur.

Sur maquette, sur simulateur, ou avec un logiciel de simulation :

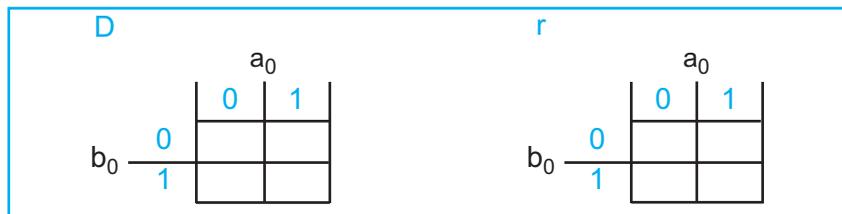
1- Câbler le logigramme suivant :



2- Compléter la table de vérité correspondante :

a_0	b_0	r	D
0	0		
0	1		
1	1		
1	0		

3- Retrouver les équations de D et r :



Equation de D

$D =$ _____

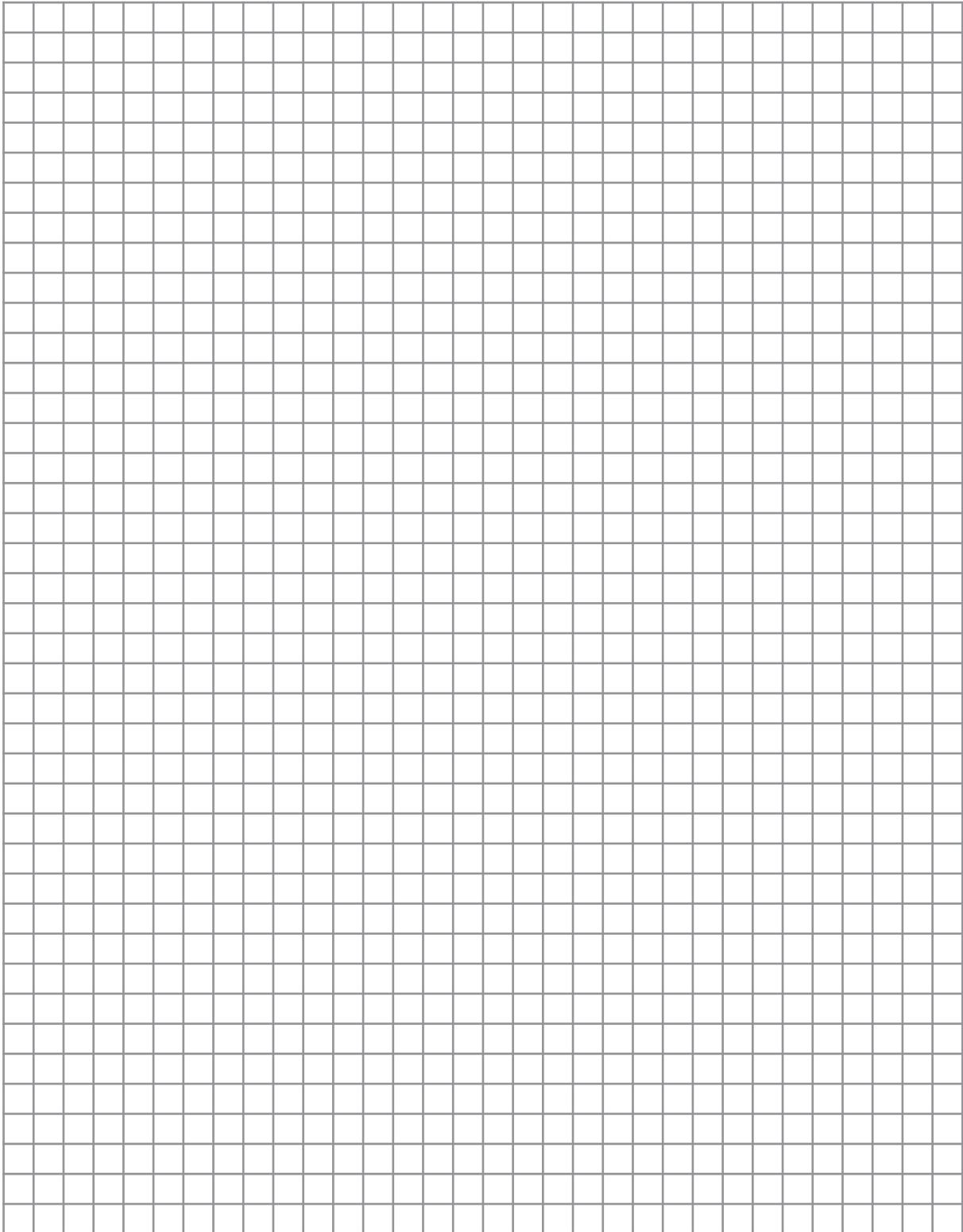
Equation de r

$r =$ _____

4- Conclure sur les valeurs de D et r en fonction de a_0 et b_0 : _____



- 5- Proposer une solution à base de portes logiques universelles **NAND** ou **NOR** (question facultative).



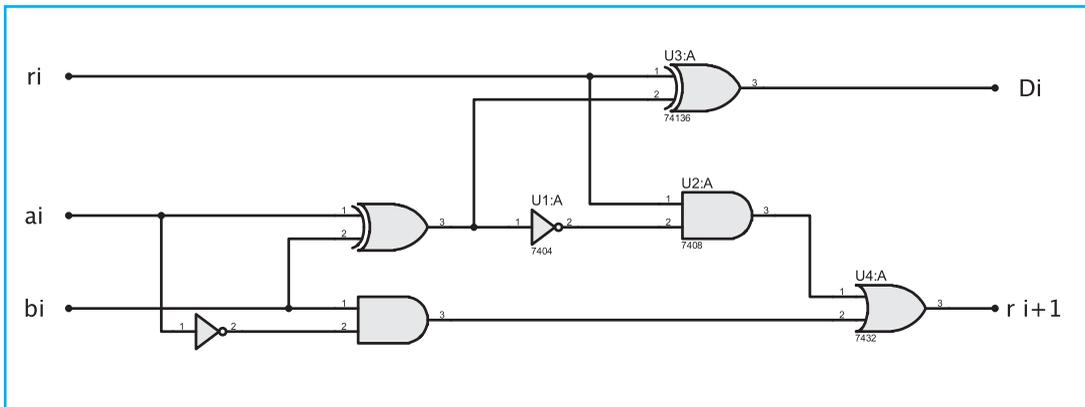


Activité 2

Il s'agit de cabler ou simuler le fonctionnement d'un circuit à base de cellules logiques, de tracer sa table de fonctionnement et de chercher les équations des sorties en vue de vérifier qu'il s'agit d'un soustracteur complet.

Sur maquette, sur simulateur, ou avec un logiciel de simulation :

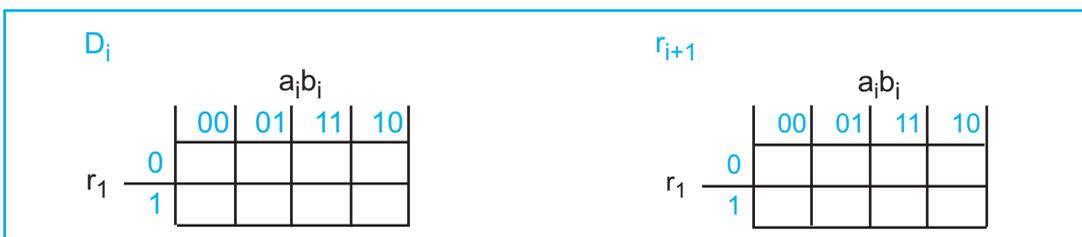
1- câbler le logigramme suivant :



2- Compléter la table de vérité correspondante :

a_i	b_i	r_i	r_{i+1}	D_i
0	0	0		
0	0	1		
0	1	1		
0	1	0		
1	1	0		
1	1	1		
1	0	1		
1	0	0		

3- Retrouver les équations de D_i et r_{i+1} :





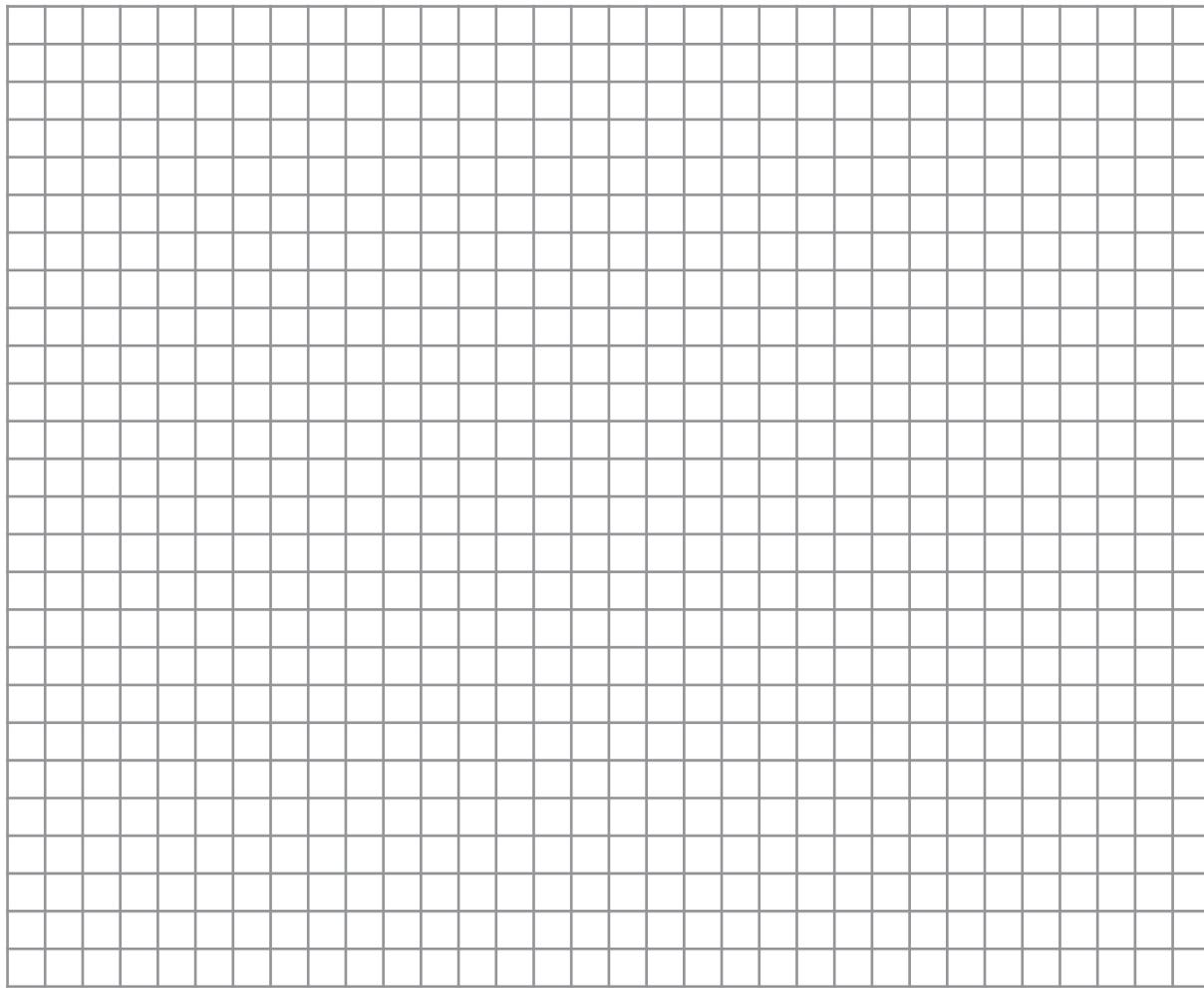
Equation de D_i	Equation de r_{i+1}
$D_i =$	$r_{i+1} =$
.....
.....
.....
.....

4- Conclure sur les valeurs de D_i et r_{i+1} en fonction de a_i , b_i et r_i :

.....

.....

5- Proposer une solution à base de portes logiques universelles **NAND** ou **NOR** (question facultative).

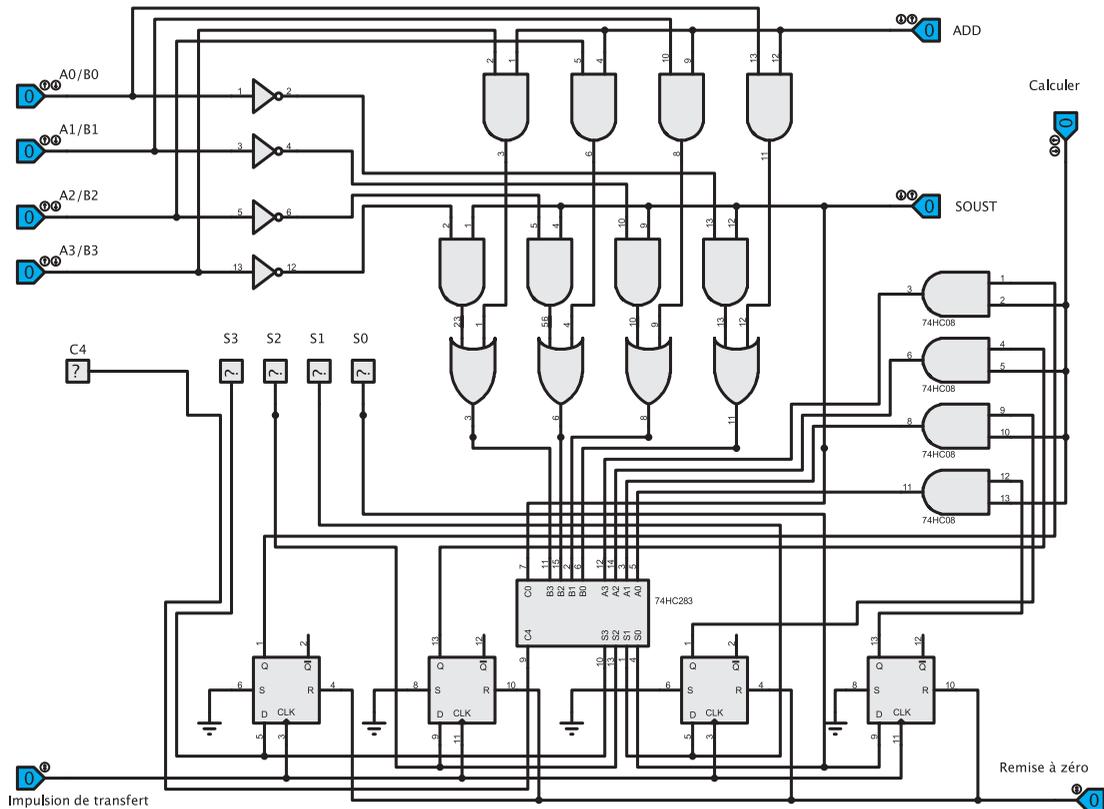




Activité 3

Il s'agit de cabler ou simuler le fonctionnement d'un circuit **Additionneur/Soustracteur** parallèle dans la notation en complément à 2 et à découvrir les périphériques de ce circuit.

Le schéma structurel suivant représente un additionneur/Soustracteur parallèle avec ses périphériques.



Légende :

- ✓ ----- : Bascule D
- ✓ 74HC283 : -----
- ✓ ----- : Cellule (OR / OU)
- ✓ ----- : Inverseur

- ✓ Ai / Bi : -----
- ✓ Si : -----
- ✓ ----- : Cellule (AND / ET)



- 1- Par analogie à l'activité 5 du TP A1-1, énoncer les différentes étapes à suivre pour tirer profit de ce circuit.

A large, empty grid of small squares, intended for the student to write their answer to the question above. The grid is composed of 20 columns and 30 rows of squares.

TP A2-1

Le Comparateur Logique

- Objectifs Spécifiques :

OS A2-1 - Résoudre un problème de logique combinatoire.

OS A2-2 - Mettre en oeuvre un circuit combinatoire.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Poste PC
- ✓ Logiciel de simulation
- ✓ Matériel d'essai en électronique numérique
- ✓ Simulateur logique
- ✓ Maquette.

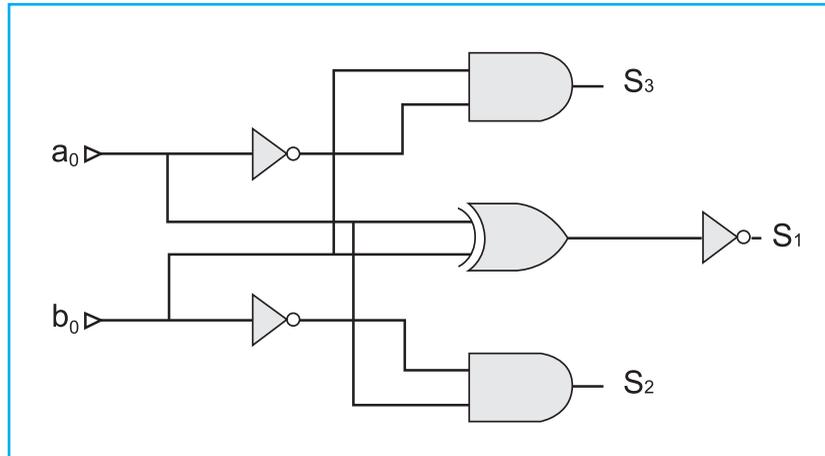


Activité 1

cette activité consiste à comparer deux nombres binaires à un bit chacun ($A : a_0$) et ($B : b_0$) et de signaler à la sortie si $A < B$, $A > B$ ou $A = B$ au moyen des sorties respectives : S_3 , S_2 et S_1 .

Sur maquette, sur simulateur, ou avec un logiciel de simulation :

1- Câbler le logigramme suivant :



2- Compléter la table de vérité correspondante :

a_0	b_0	S_1	S_2	S_3
0	0			
0	1			
1	1			
1	0			

3- Retrouver les équations de S_1 , S_2 et S_3 :

<p>S_1</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td colspan="2">a_0</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td rowspan="2">b_0</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td></td> </tr> </table>		a_0			0	1	b_0	0		1		<p>S_2</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td colspan="2">a_0</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td rowspan="2">b_0</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td></td> </tr> </table>		a_0			0	1	b_0	0		1		<p>S_3</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td colspan="2">a_0</td> </tr> <tr> <td></td> <td>0</td> <td>1</td> </tr> <tr> <td rowspan="2">b_0</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td></td> </tr> </table>		a_0			0	1	b_0	0		1	
	a_0																																		
	0	1																																	
b_0	0																																		
	1																																		
	a_0																																		
	0	1																																	
b_0	0																																		
	1																																		
	a_0																																		
	0	1																																	
b_0	0																																		
	1																																		

$S_1 =$ Equation de S_1

$S_2 =$ Equation de S_2

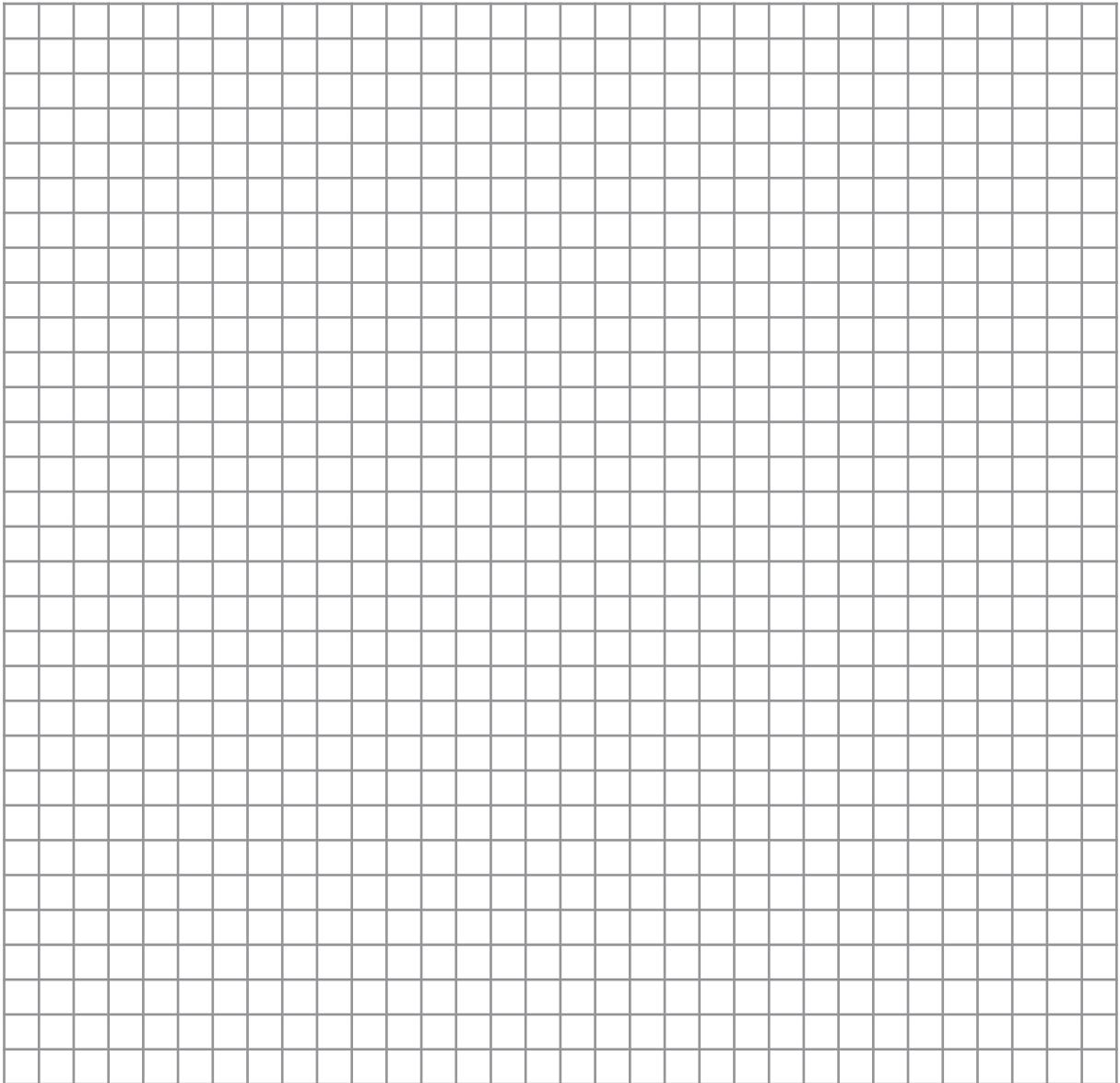


Equation de S3

S3 =
.....
.....

4- Conclure sur les valeurs de S1, S2 et S3 en fonction de a0 et b0 :
.....
.....

5- Proposer une solution à base de portes logiques universelles NAND ou NOR (question facultative).





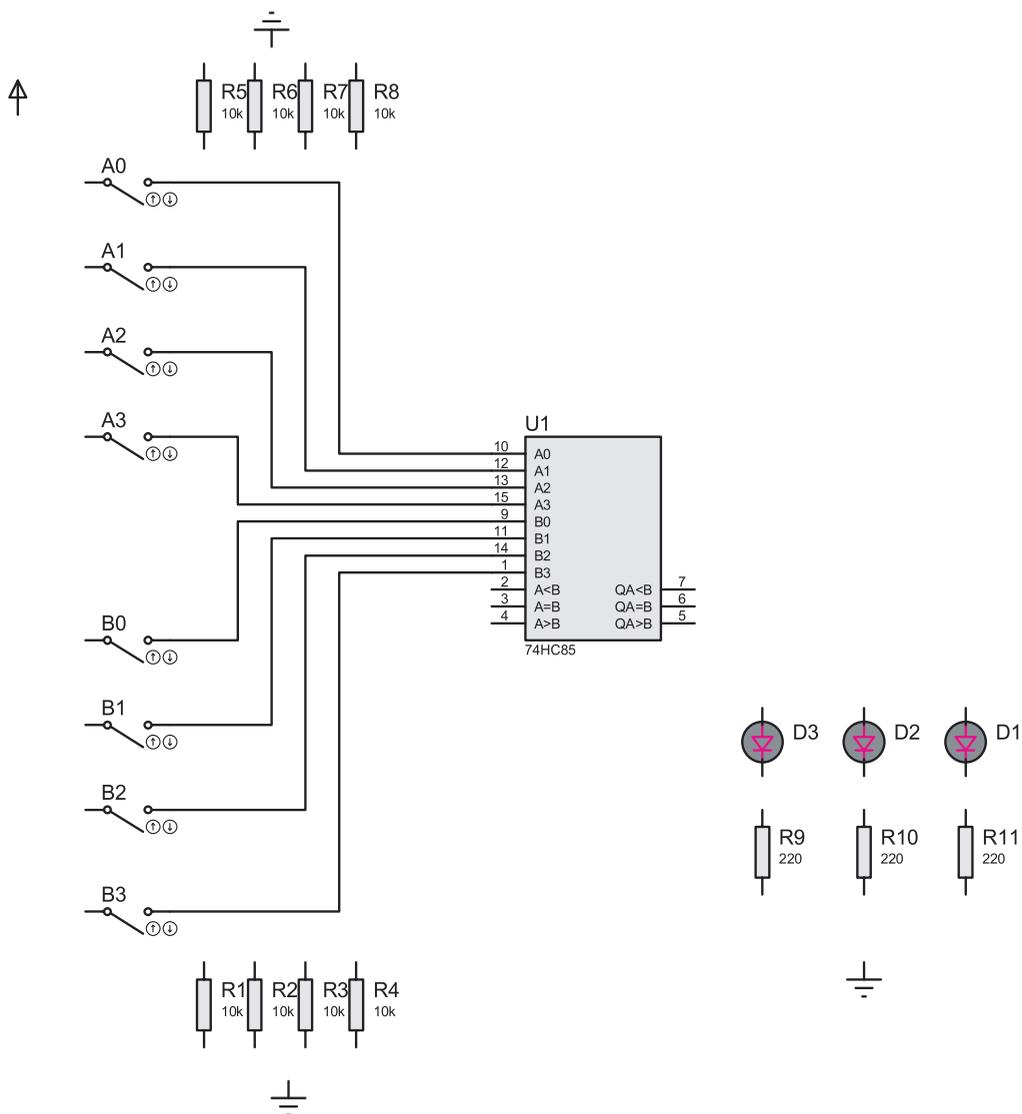
Activité 2

MISE EN ŒUVRE DES COMPAREURS INTEGRES

Dans l'activité qui suit, on se propose d'étudier un comparateur de la famille **74HC85 / 74HCT85** ; ce comparateur possède deux entrées à **4 bits** :

- $A = (A_3A_2A_1A_0)$.
- $B = (B_3B_2B_1B_0)$.

1- Compléter le logigramme correspondant :



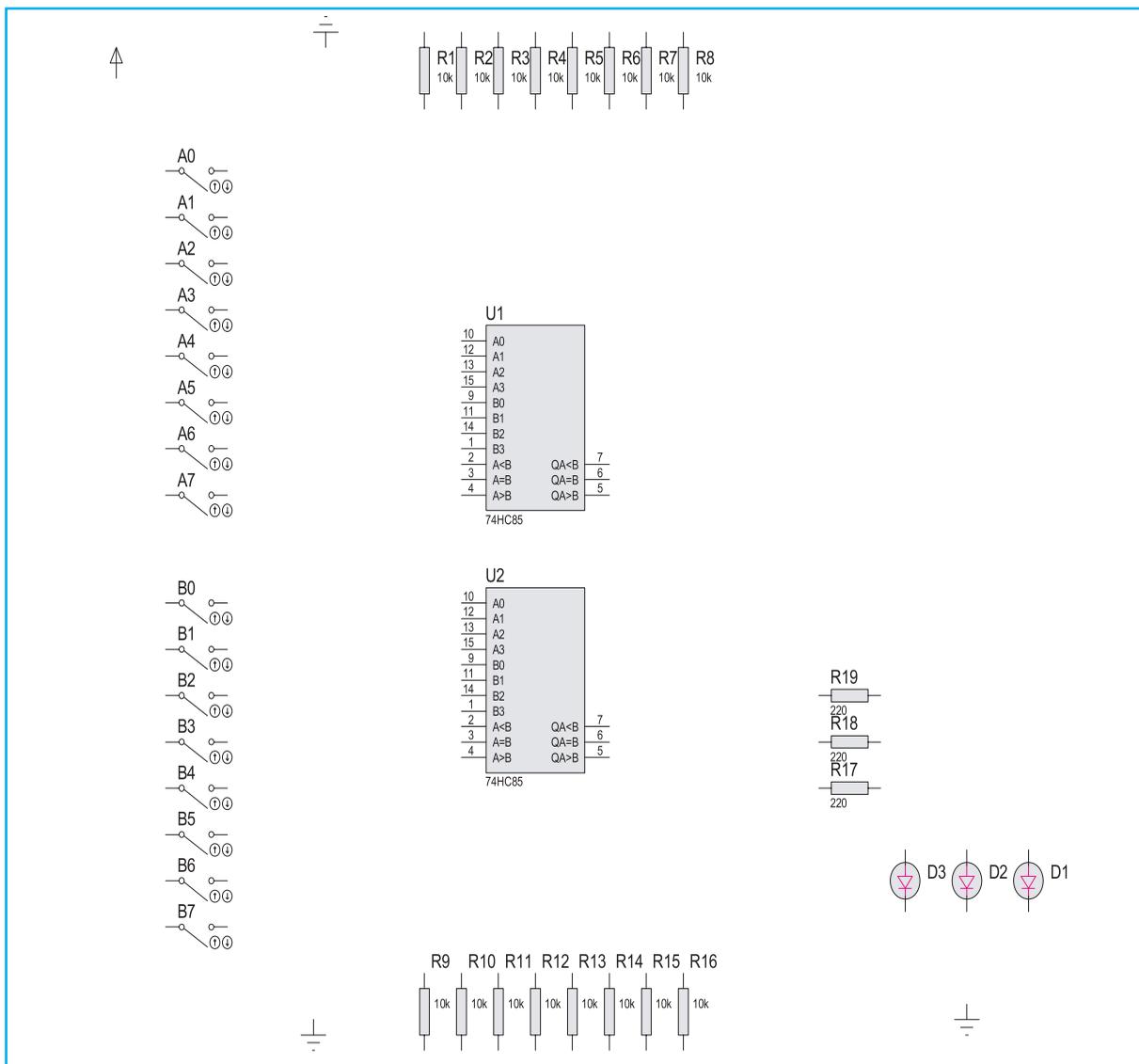


Activité 3

MISE EN CASCADE DE COMPARATEURS INTEGRES

Cette activité consiste à mettre en cascade deux comparateurs intégrés de la famille **74XXX85** en vue de comparer deux nombres binaires de 8 bits chacun.

1- Compléter le logigramme relatif à un **comparateur** de deux nombres à **8 bits** (**A** : **A₇A₆A₅A₄A₃A₂A₁A₀** et **B** : **B₇B₆B₅B₄B₃B₂B₁B₀**), utiliser des circuits de la famille **74HCT85 / 74HC85** ou équivalent.



2- Sur maquette, sur simulateur ou avec un logiciel de simulation :

- a. Câbler le logigramme proposé.
- b. Vérifier son fonctionnement.

TP A2-2

Unité Arithmétique et Logique (UAL-ALU)

- Objectifs Spécifiques :

- OS A1-1 - Exécuter en binaire une opération arithmétique de base
- OS A2-2 - Mettre en oeuvre un circuit arithmétique.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

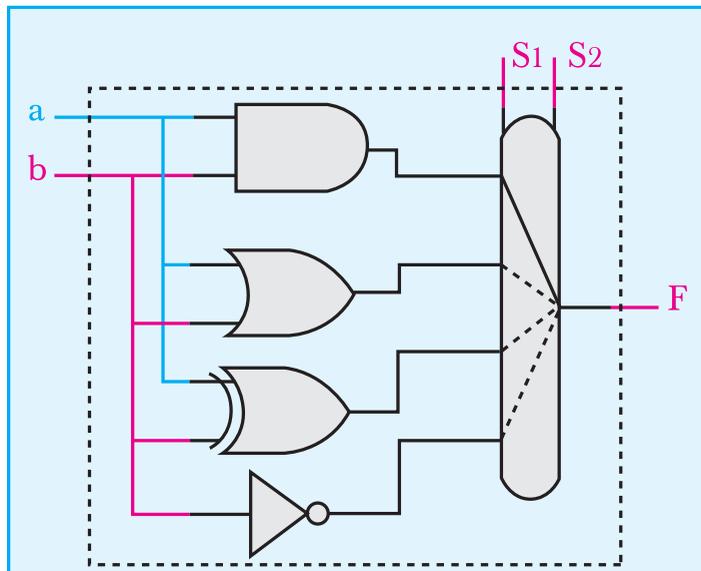
- ✓ Poste PC
- ✓ Logiciel de simulation
- ✓ Matériel d'essai en électronique numérique
- ✓ Simulateur logique
- ✓ Maquette.



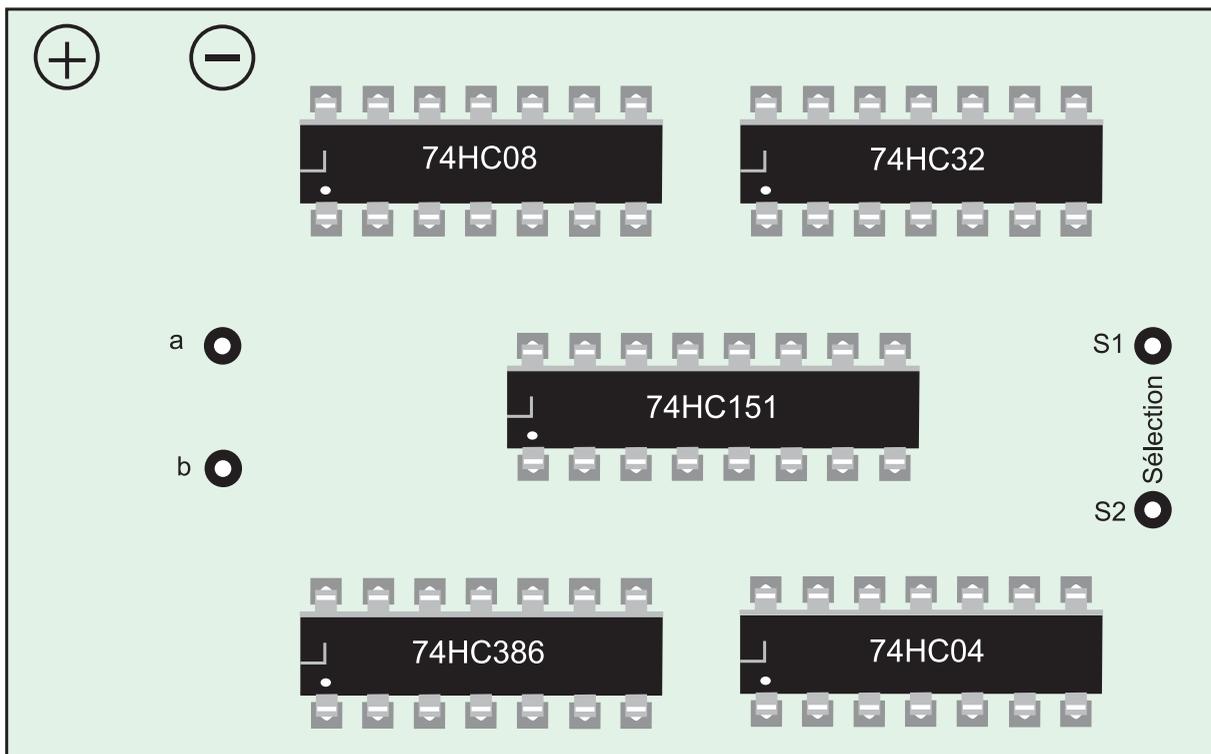
Activité 1

Dans cette activité on compte réaliser une UL élémentaire réalisant quelques opérations logiques.

Soit le logigramme suivant :



1- Câbler la platine ci-après, conformément au logigramme précédent.





2- En utilisant le datasheet relatif au circuit 74HC151 " fourni par le professeur ou récupéré sur Internet ", compléter la table de fonctionnement suivante :

Sélection		Fonction
S ₂	S ₁	
0	0	
0	1	
1	0	
1	1	

Légende :

+ 74HC04 : -----

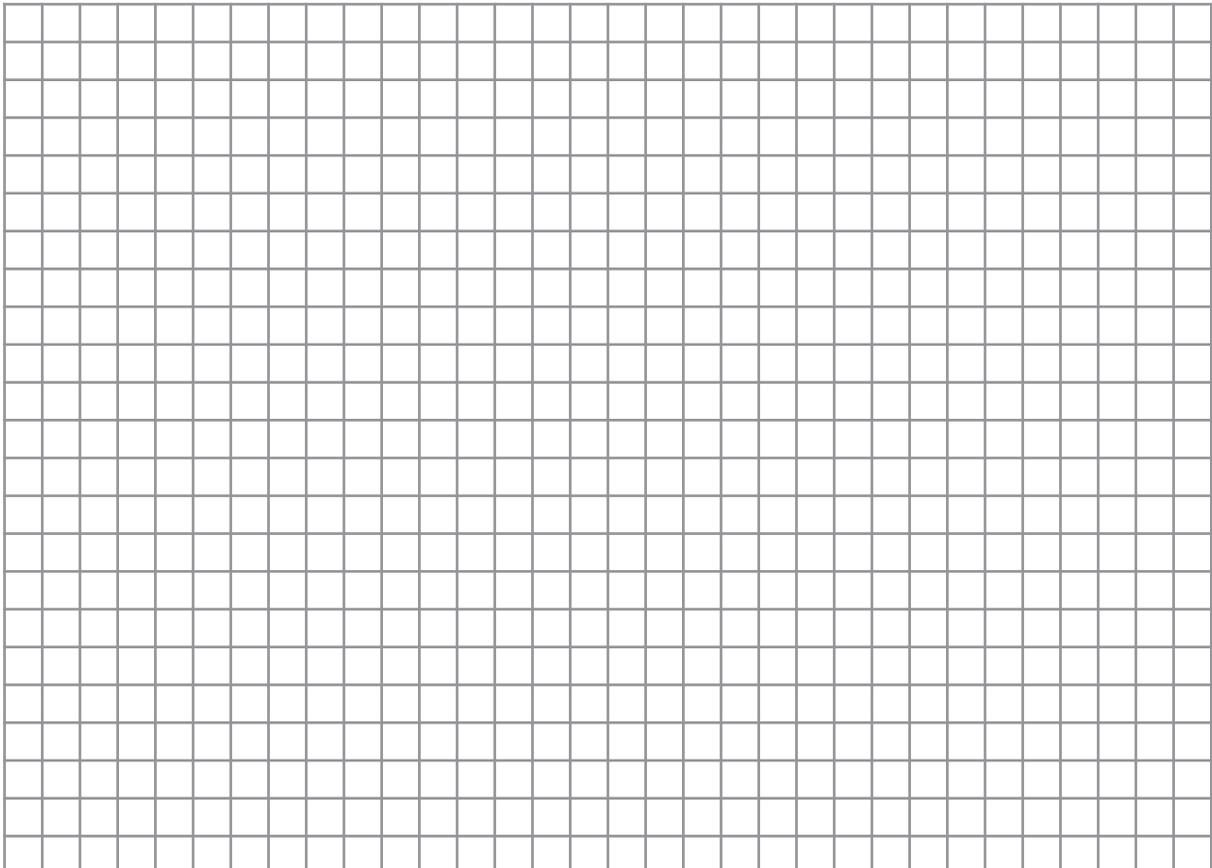
+ 74HC08 : -----

+ 74HC32 : -----

+ 74HC386 : -----

+ 74HC151 : -----

3- Conclure :





Activité 2

Mise en oeuvre d'une UAL de la série 74XX

Parmi ces circuits on cite, **74S181**, **74LS181**, **74HC181**, **74LS381** et **74S381**.

- 1- A partir du document constructeur relatif au composant disponible au laboratoire, énumérer les principales fonctions réalisées par ce composant.

- 2- Sur maquette ou avec le logiciel de simulation, tester le composant.

TP A3-1

Les Compteurs

- Objectifs Spécifiques :

OS A3-1 - Réaliser des applications à base de bascules

OS A3-2 - Choisir et mettre en oeuvre un compteur / décompteur.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Poste PC
- ✓ Logiciel de simulation logique installé (ORCAD)
- ✓ Matériel d'essai en électronique numérique
- ✓ Simulateur logique.



Activité 1

Pour étudier le fonctionnement du compteur de gélules, on utilise une maquette d'essai qui permettra de simuler l'opération de comptage. On demande de remplir la table de comptage relative au compteur utilisé sur cette maquette et d'en déduire le cycle de fonctionnement.

La maquette d'étude est une carte électronique réalisée autour d'un compteur intégré référencé CD 4510.

- 1- Les broches 3, 4, 12 et 13 correspondent aux entrées de prépositionnement du compteur. Vérifier que la broche 3 du compteur est reliée à la masse et que les broches 4, 12 et 13 sont reliées au + de l'alimentation.
- 2- L'entrée de comptage est issue d'un capteur optique situé sur la trajectoire de passage des gélules. On prévoit sur la maquette d'étude un bouton poussoir H qui remplace le capteur optique. Appuyer chaque fois sur le bouton poussoir H et remplir une ligne de la table de fonctionnement en suivant l'évolution des entrées/sorties du compteur (visualisées sur Leds) avant et après l'action :

Décimal	Etat n			Etat n+1					
	Q2	Q1	Q0	Q2		Q1		Q0	
0	0	0	0	0	$\mu 0$	0	$\mu 0$	1	ϵ
1									
2									
3									
4									
5									
6									
7									

- 3- En supposant que ce compteur est construit avec des bascules JK :
 - a. Rappeler le diagramme de fluence d'une bascule JK.



- b. En utilisant le diagramme de fluence de la bascule **JK**, compléter les tableaux de **KARNAUGH** relatifs aux différentes entrées de commande puis déterminer les équations de **J** et **K** de chaque bascule :



Q1Q2 \ Q3	00	01	11	10
0				
1				

J2

Q1Q2 \ Q3	00	01	11	10
0				
1				

J1

Q1Q2 \ Q3	00	01	11	10
0				
1				

J0

Q1Q2 \ Q3	00	01	11	10
0				
1				

K2

Q1Q2 \ Q3	00	01	11	10
0				
1				

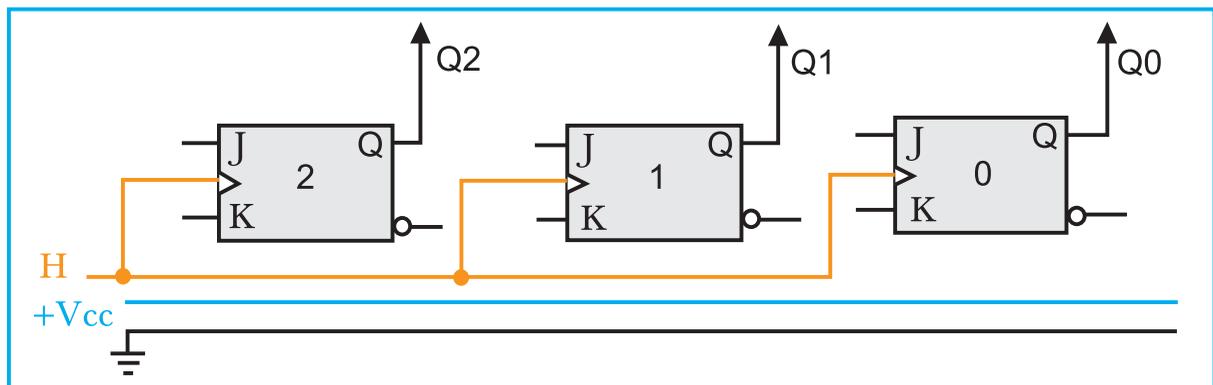
K1

Q1Q2 \ Q3	00	01	11	10
0				
1				

K0

J2 = **J1** = **J0** =
K2 = **K1** = **K0** =

4- Donner le schéma du compteur de gélules à base de bascules JK qui permet de faire le même cycle de comptage que le circuit intégré de la maquette d'étude.

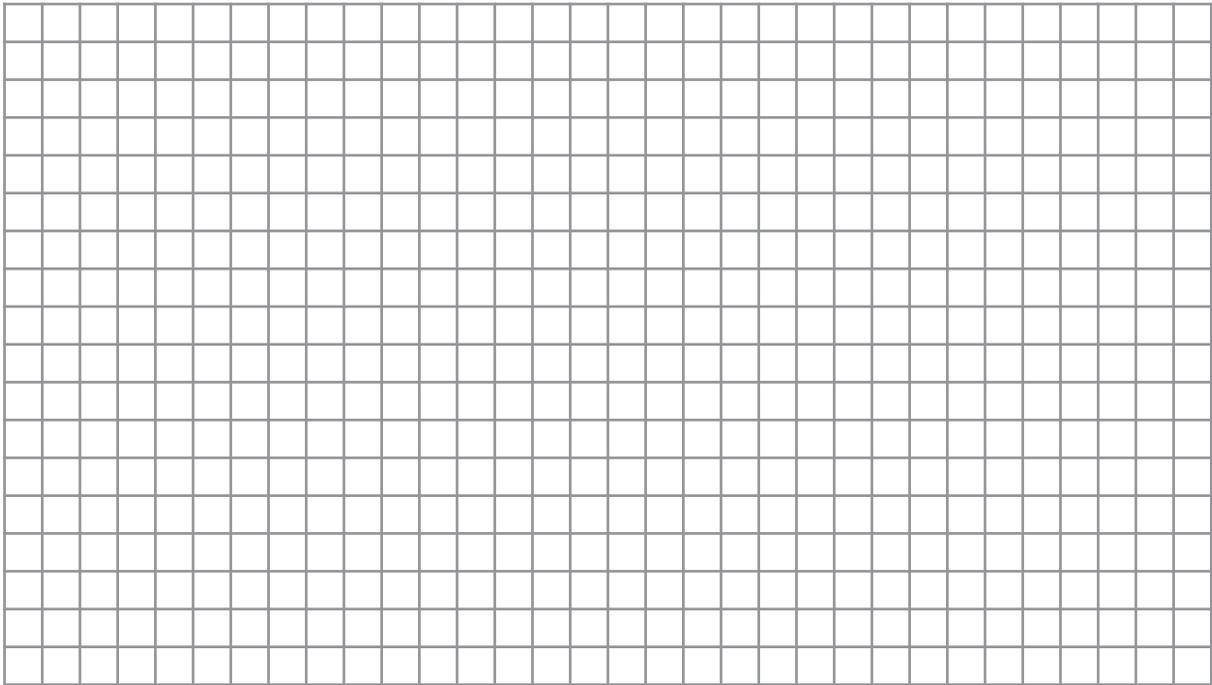




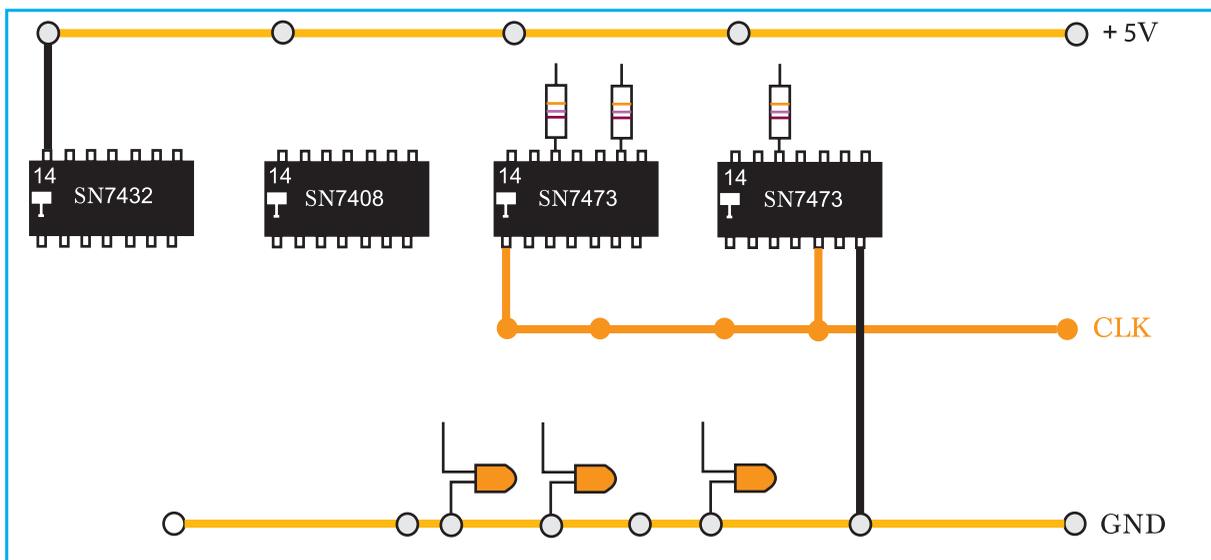
Activité 2

Il s'agit de réaliser sur plaque à essais un compteur synchrone à base de bascules JK conformément aux équations trouvées dans l'activité 1.

- 1- A partir d'une recherche INTERNET (sinon document fourni), identifier, selon les références, les différents circuits de la maquette.



- 2- A partir des équations trouvées dans l'activité 1, compléter sur la figure les liaisons qui manquent sur le circuit électronique :





- 3- Réaliser sur plaque à essais le montage de la figure précédente puis alimenter le circuit réalisé et vérifier la conformité du fonctionnement.

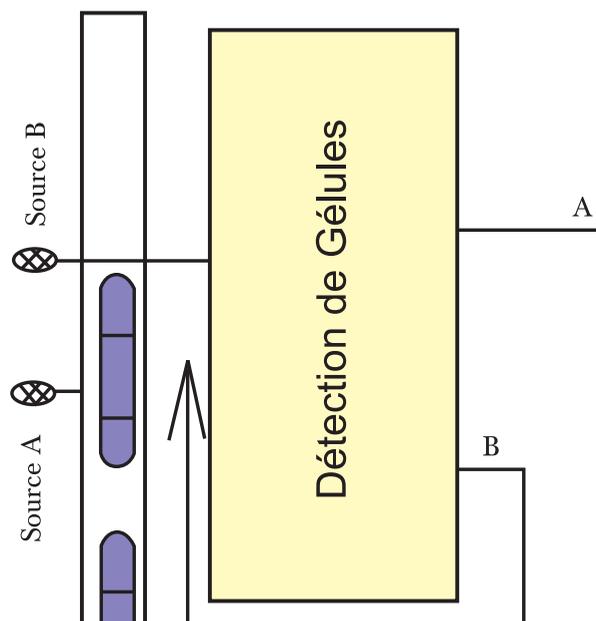
Activité 3

Il s'agit d'étudier un dispositif qui permet de gérer les défauts de comptage des gélules dans la compteuse industrielle. A cet effet on se propose de simuler sur ORCAD le fonctionnement d'un compteur / décompteur prépositionnable et de tracer le cycle de comptage pour différentes combinaisons des entrées de prépositionnement.

Le mouvement vibratoire des gélules constitue une source d'erreur importante, car une gélule, par ses mouvements d'avance /recul peut être comptée deux fois ou plus. Un dispositif de comptage/décomptage permet de palier à cette défaillance technique.

Ce dispositif se compose de :

- deux sources lumineuses espacées par une distance très faible, très nettement inférieure à la dimension des gélules, mais suffisante pour qu'il n'y ait aucune interférence lumineuse entre les deux capteurs,
- deux phototransistors destinés à détecter la présence ou l'absence de gélules entre chaque source et son phototransistor,
- un ensemble de mise en forme des signaux en provenance des phototransistors,
- un système de comptage/décomptage permettant de connaître le nombre exact de pièces emballées, même si une pièce est passée plusieurs fois, en marche avant puis en marche arrière sur le tapis roulant.



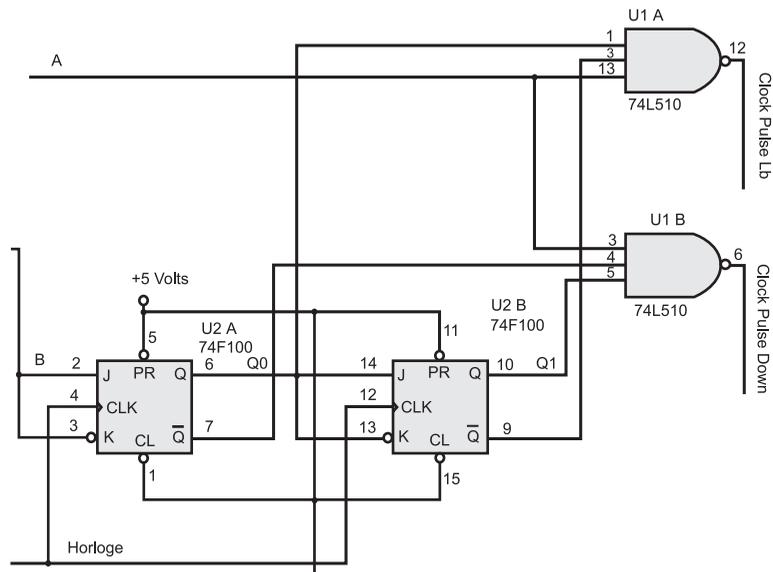


Principe de fonctionnement

Lorsqu'**aucune gélule** ne coupe le faisceau issu de la source lumineuse A, **le signal A** en sortie du circuit de détection et de mise en forme est **à zéro**. Au contraire, si **une gélule coupe** ce faisceau, alors le signal **A est à un**.

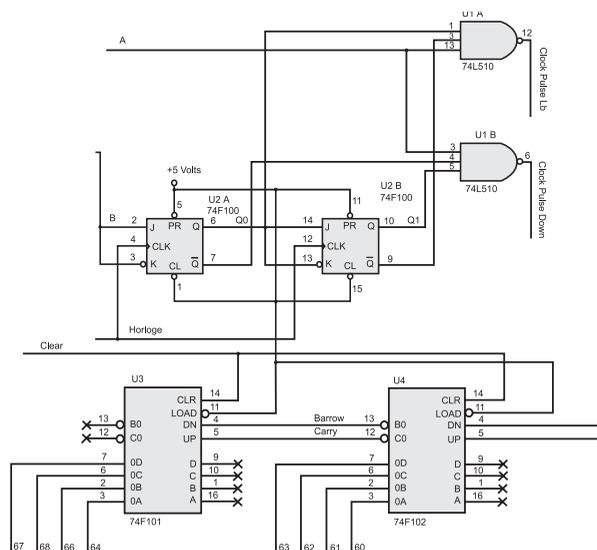
Le signal **B** fonctionne de façon **inverse**. Il est à 1 si aucune pièce ne coupe le faisceau issu de la source lumineuse B, et à 0 si une pièce coupe ce faisceau.

Le signal B est utilisé en entrée d'un ensemble de deux bascules JK de référence 74F109 (technologie TTL Fast), dont les entrées sont en fait J et K (voir le schéma ci-dessous). Une horloge H à 1 Mhz sert de signal de chargement pour ces deux bascules.



Les sorties **Q0**, **/Q0**, **Q1** et **/Q1** de ces bascules sont associées au signal A en entrées de deux portes NAND de type 74LS10, pour réaliser les deux signaux de comptage / décomptage.

Clock Pulse Up et **Clock Pulse Down** de l'ensemble réalisé avec les deux compteurs synchrones **74F192** (schéma ci-dessous et schéma complet ci-joint).





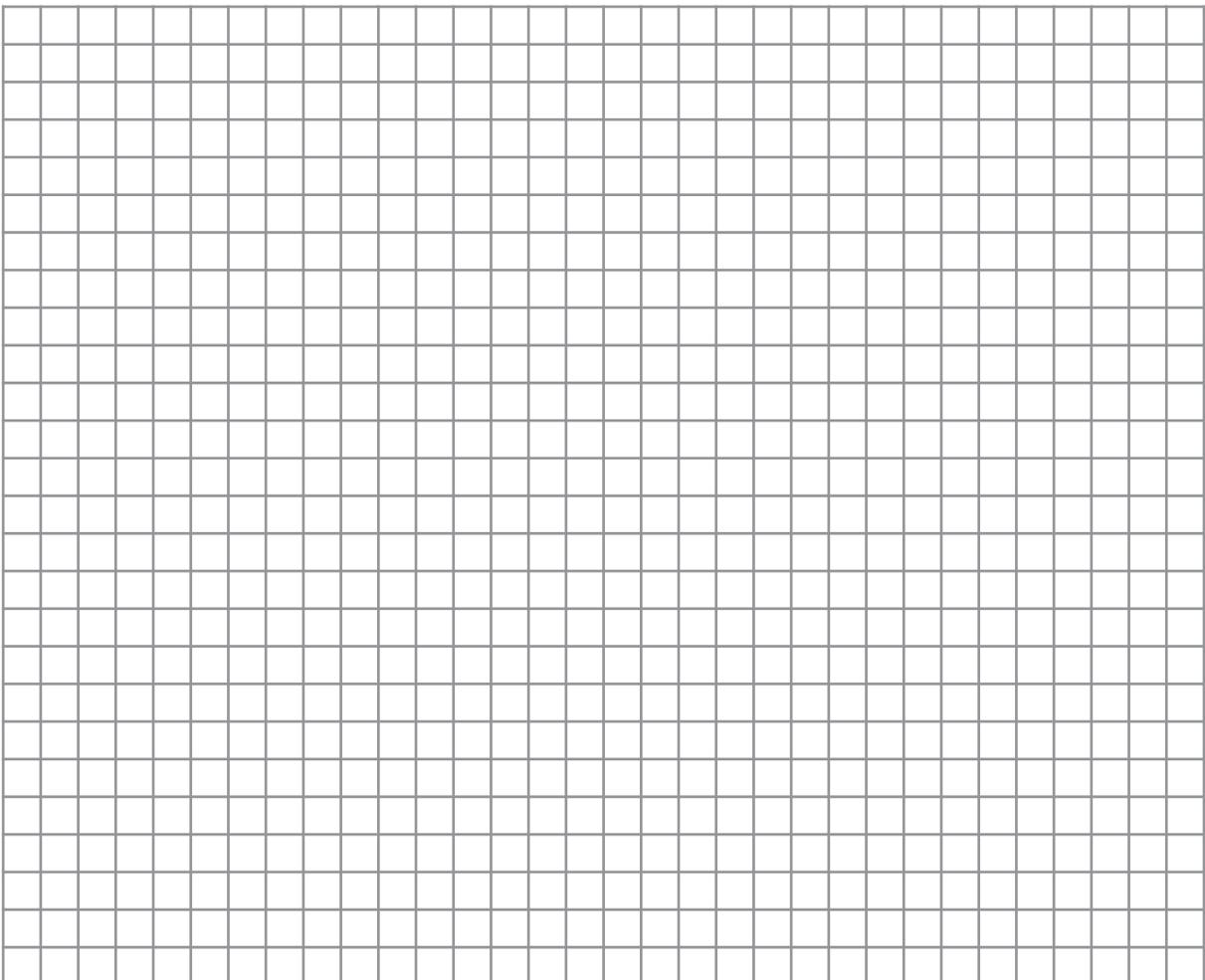
Génération des signaux de comptage

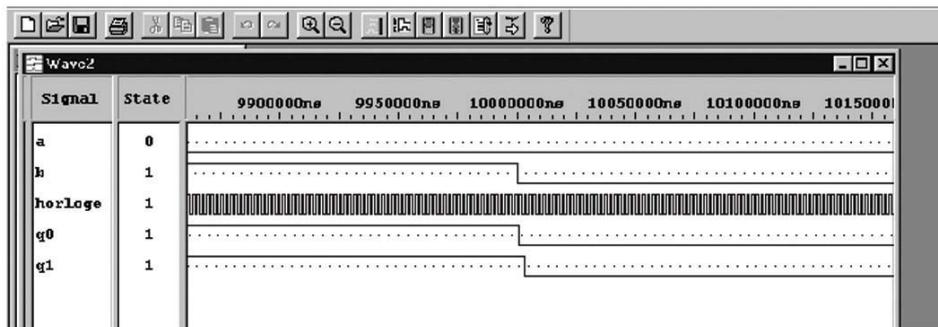
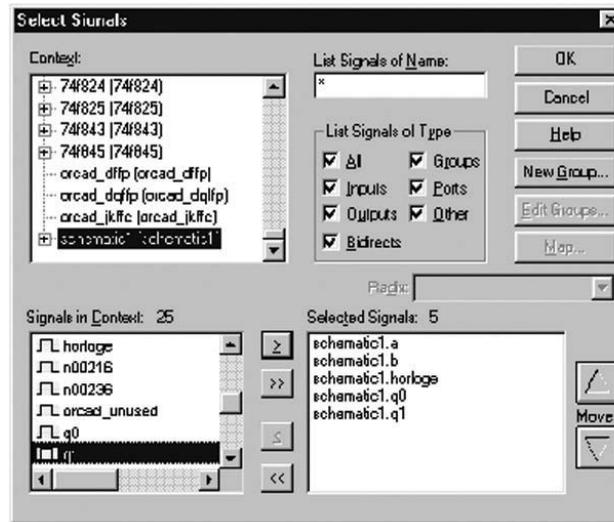
Etude du principe de fonctionnement des bascules JK.

- Rappeler le principe de fonctionnement d'une bascule JK.
- En déduire le mode de fonctionnement des bascules 74F109 utilisées dans le montage.
- Si B est un signal rectangulaire périodique de période $T = 100$ ms, représente par un chronogramme les signaux B, Horloge, Q0 et Q1.
- Après avoir lancé le logiciel Simulate, charger le fichier 'TP2' avec ses stimulus, puis demander à tracer une nouvelle fenêtre de signaux et visualiser dans cette fenêtre les signaux a, b, horloge, q0 et q1 (voir la fenêtre de commande à la page ci-après).
- Lancer alors la simulation (Commande Run - Start) sur une durée de 100 ms (soit 100000000 ns), et examiner le fonctionnement des deux bascules lorsqu'une pièce se présente devant la source B (ce qui arrive aux environs de 10 ms).

Expliquer dans le détail l'allure des différents signaux, en justifiant le raisonnement avec l'étude du fonctionnement de la bascule JK faite précédemment.

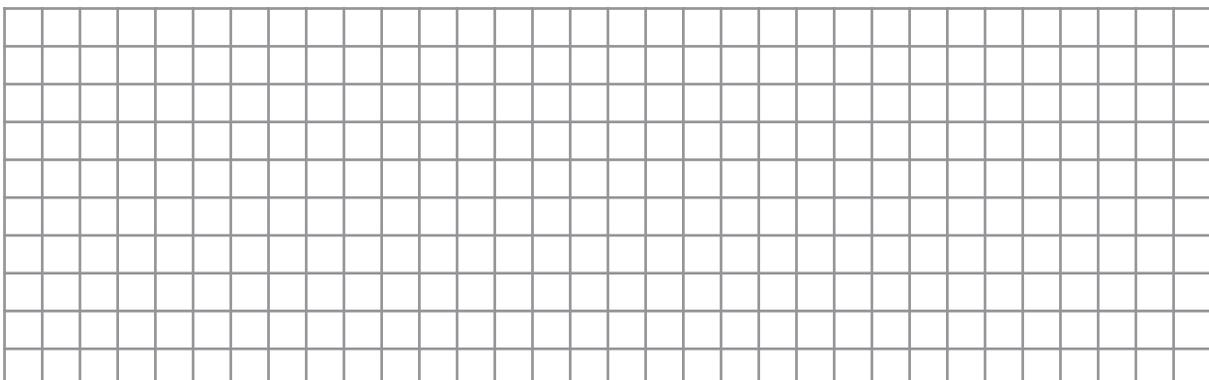
On conseille d'utiliser pour cela la loupe du logiciel afin d'examiner avec précision le fonctionnement dans le temps des chronogrammes.





Génération des signaux de comptage

- A l'aide de la fonction Trace - Edit, ajouter aux signaux précédents les deux signaux Clock Pulse Up et Clock Pulse Down.
- Relancer la simulation (Run - Restart puis Start) sur une même durée de 100 ms, et examiner les deux signaux Clock Pulse Up et Down. Justifier clairement leur allure, et en déduire le fonctionnement des compteurs.
- A l'examen des signaux A et B dire si une ou plusieurs gélules se déplacent devant les sources lumineuses, et dans quel sens.
- Le fonctionnement attendu des compteurs est-il conforme à ce déplacement de gélules ?



TP A4-1

Les Automates Programmables Industriels

- Objectifs Spécifiques :

OS A3-3 - Décrire le fonctionnement d'un système automatisé à l'aide de l'outil GRAFCET.

OS A4-1 - Identifier les éléments de dialogue d'un système automatisé piloté par API.

OS A4-2 - Traduire un GRAFCET en langage automate.

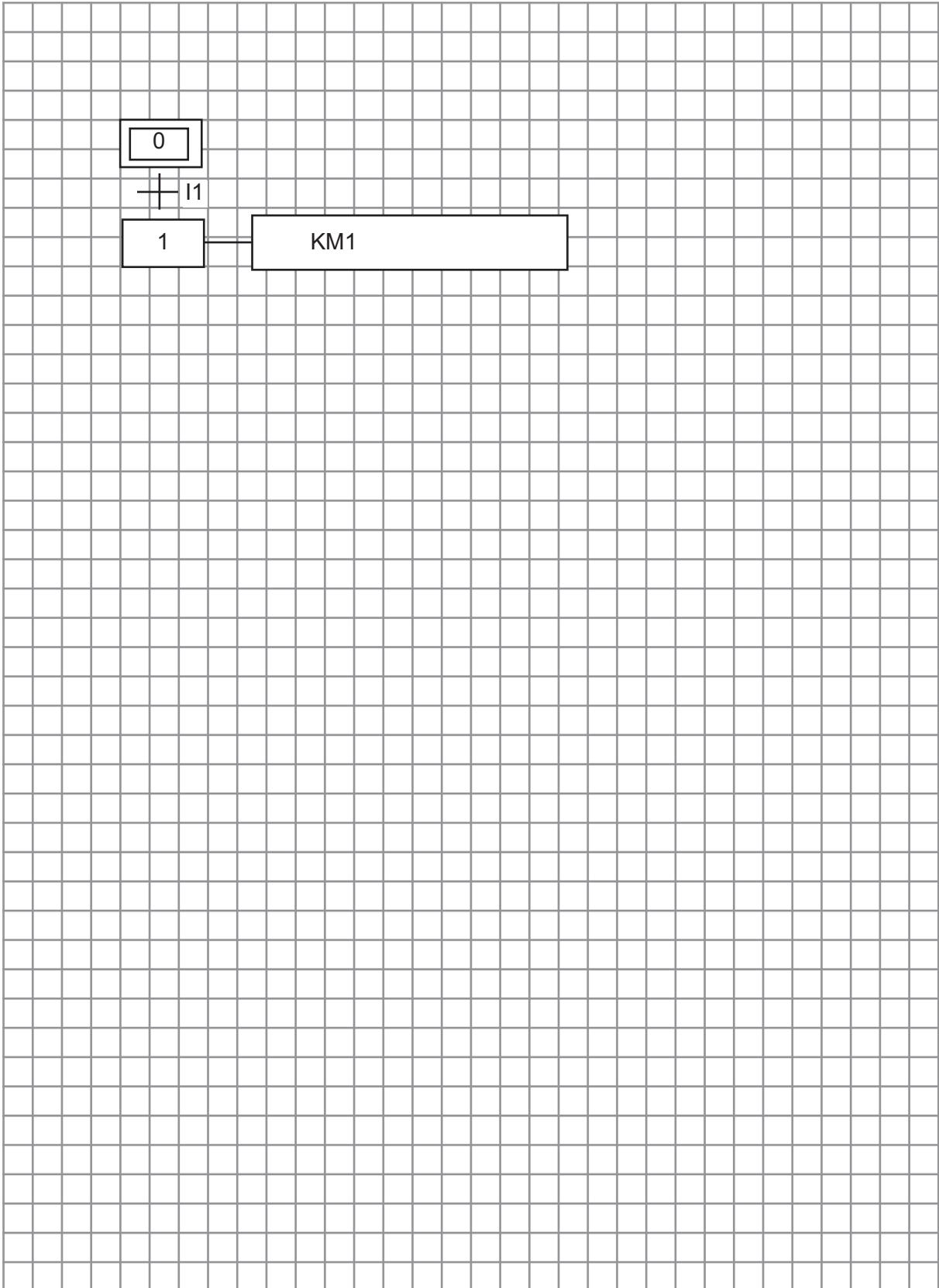
OS A4-2 - Ecrire ou compléter un programme et l'implanter sur un API.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Poste PC
- ✓ Logiciels des automates installés (PL7 et AKL)
- ✓ Automate Programmable TSX ou AEG
- ✓ Un système électropneumatique ou électrique
- ✓ Un moteur asynchrone triphasé avec démarreur étoile-triangle câblé.



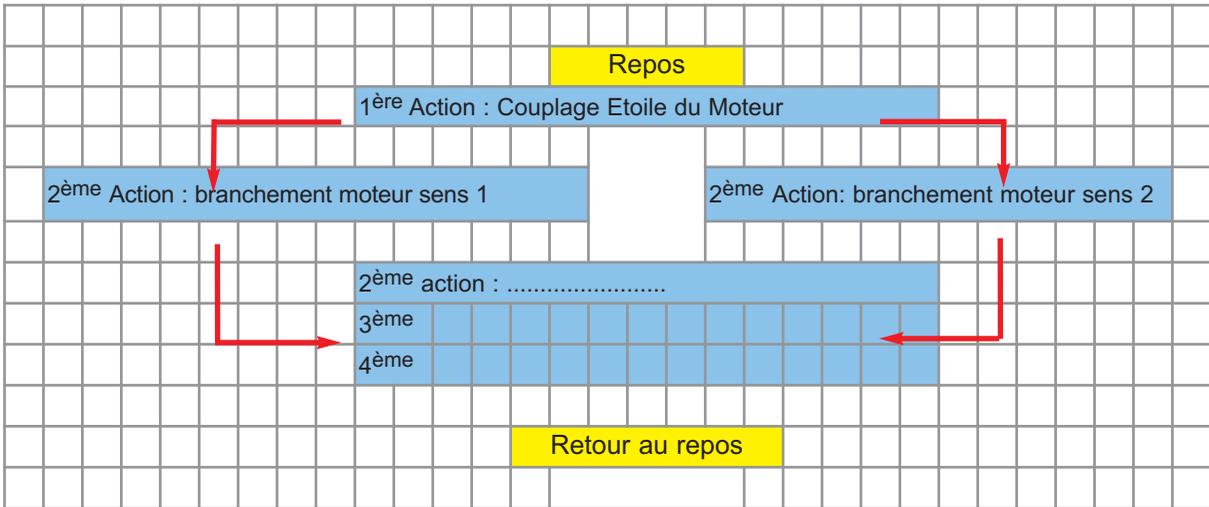


3- On désire modifier le mode de fonctionnement du système de façon que le moteur puisse tourner dans le sens inverse pour des opérations de débouillage parfois indispensable.

A partir de la position de repos on peut soit :

- démarrer le moteur dans le sens 1 en appuyant sur un bouton I11
- démarrer le moteur dans le sens 2 en appuyant sur un bouton I12
- l'arrêt dans les deux cas se fait par appui sur I0.

a. Proposer une liste des actions réalisées dans ce cas par le système.



b. Que devient le nombre de modules étapes du séquenceur de pilotage pour satisfaire cette nouvelle condition ? Justifier votre réponse.

c. Donner une estimation approximative du temps nécessaire pour recabler le séquenceur et remettre le système de démarrage en marche. Commenter.

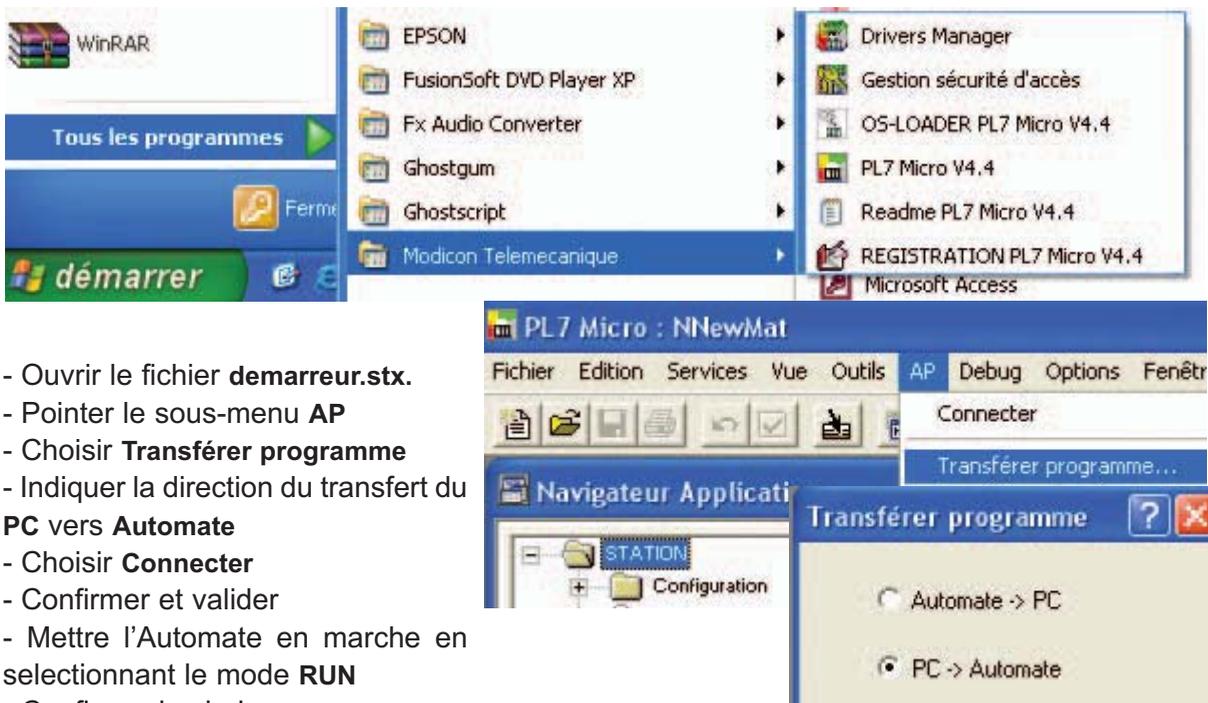
Activité 2

Il s'agit de faire tourner le même système avec un automate programmable, d'observer son fonctionnement et d'identifier les nouveaux éléments de l'installation

- Débrancher les liaisons entre les entrées du séquenceur et le système et les relier aux entrées de la nouvelle boîte de commande appelée momentanément **API**.
- Déconnecter les sorties du séquenceur de pilotage et les connecter avec les premières sorties de la boîte **API**.
- Vérifier attentivement tous les raccordements et mettre l'ensemble sous tension.
- Appuyer sur le bouton **I1**, observer le comportement du système et comparer avec les résultats de l'activité 1.
- Connecter l'**Automate** au port **DB9** du **PC** en utilisant le câble approprié.

Puisque la boîte qui a été nommée **API** a pu remplacer tout l'**automatisme** précédent sans câblage intérieur, elle prendra le nom d'**Automate**.

- Lancer le logiciel **PL7 Micro V4.4** de **Modicon Télémechanique** (ou le logiciel spécifique à l'**Automate AKL**).



- Ouvrir le fichier **demarreur.stx**.
- Pointer le sous-menu **AP**
- Choisir **Transférer programme**
- Indiquer la direction du transfert du **PC** vers **Automate**
- Choisir **Connecter**
- Confirmer et valider
- Mettre l'Automate en marche en sélectionnant le mode **RUN**
- Confirmer le choix.



3- Sachant que les notations :

Entrées Booléennes spécifiques aux :

* Automates TSX %I1.0 %I1.15

* Automates AEG I1. I 24

Sorties boléennes spécifiques aux

* Automates TSX %Q2.0 %Q2.11

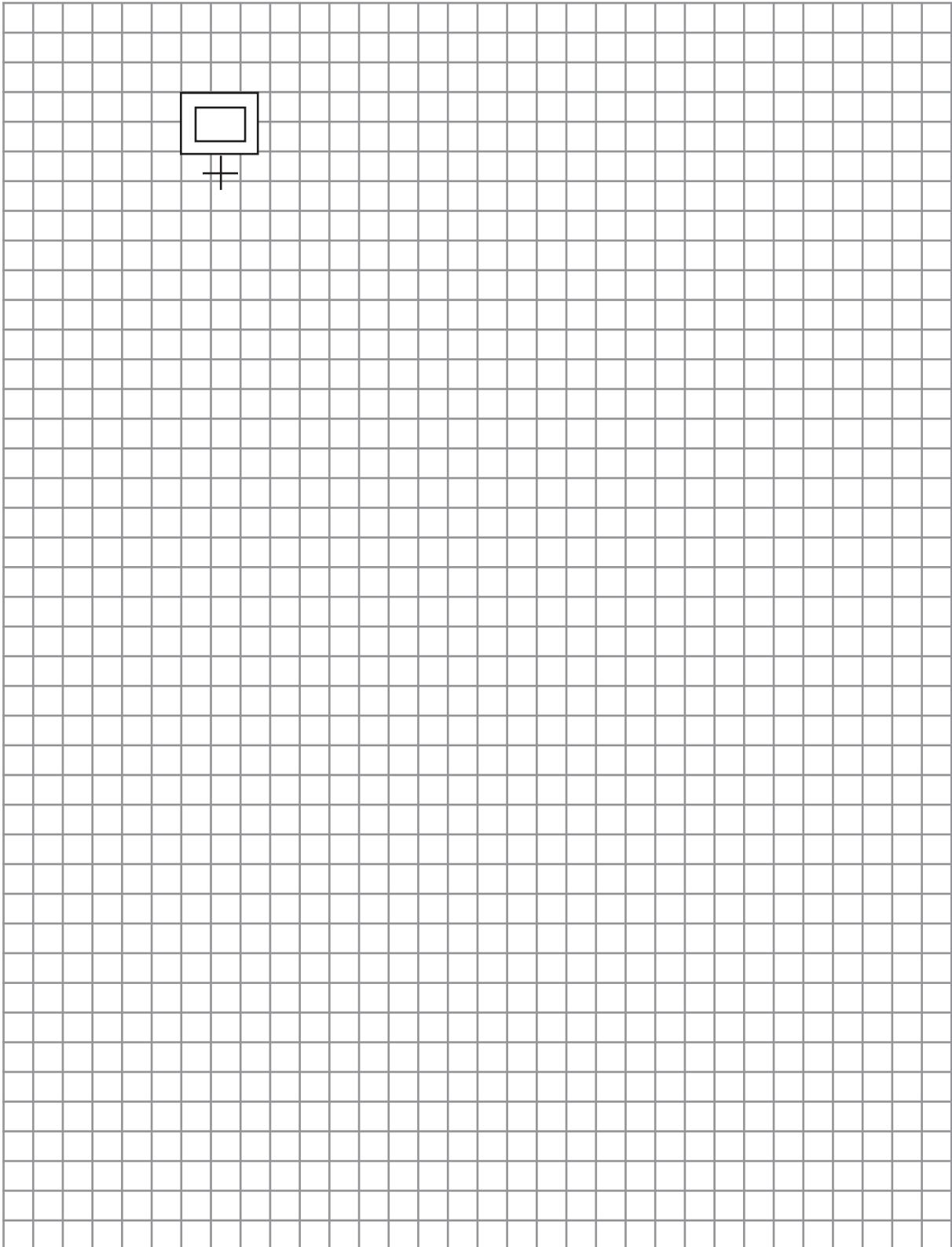
* Automates AEG Q2. Q16

tracer le GRAFCET codé Automate relatif au nouveau cycle de fonctionnement tout en respectant le tableau d'affectation donné ci-dessous.

Liste des informations				Liste des actions			
Label	Désignation	TSX	AEG	Label	Désignation	TSX	AEG
I ₁	mise en marche sens 1	%I1.1	I1	KM11	Contacteur ligne sens1	%Q2.1	Q1
I ₂	mise en marche sens 2	%I1.2	I2	KM12	Contacteur ligne sens2	%Q2.2	Q2
I ₀	bouton d'arrêt	%I10	I3	KM3	Contacteur "Etoile"	%Q2.3	Q2
t	contact temporisé	%TM0.Q	T	KM4	Contacteur "Triangle"	%Q2.4	Q2
				T	Bobine Temporisateur	%TM0	T



GRAFCET codé Automate

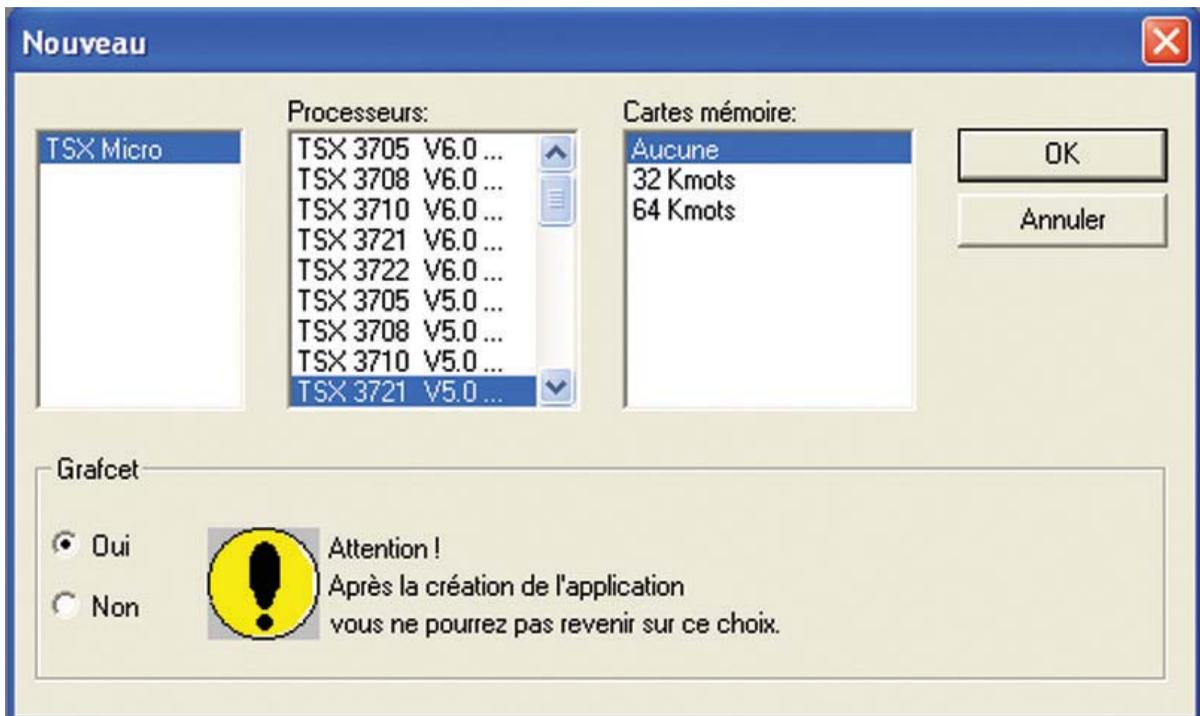


Activité 3

Il s'agit de configurer correctement un automate programmable, en se référant aux documents constructeur et d'identifier les éléments du système raccordés à ses entrées/sorties.

Pour pouvoir exploiter l'Automate Programmable dans une application industrielle, il faut configurer ses différents constituants afin d'être reconnus par le processeur. La tâche suivante consiste à suivre une démarche exploratrice réalisant la dite opération de configuration.

- lancer le logiciel PL7.
- dans le menu "Fichier" cliquer sur "Nouveau" pour créer un nouveau fichier.



- Faire le choix du type d'Automate et cocher l'option "oui" si dans le fichier on aura à utiliser le GRAFCET pour programmer la commande du système à piloter puis valider, on obtient l'écran ci-contre
- en cliquant sur "Configuration" on a accès aux menus de configuration matérielle et logicielle.

**Configuration matérielle de l'Automate :**

- pointer le module 1 et cliquer 2 fois puis configurer le module d'E/S bloc 1 et 2 en choisissant Tout ou rien et TSX DMZ28DR.

Activité 4

Il s'agit de traduire un GRAFCET codé automate en un programme Liste d'Instructions permettant de piloter un système de commande d'un moteur asynchrone dans les deux sens de marche (**sur Automate TSX**).

A partir du **GRAFCET codé Automate** tracé dans l'activité précédente, écrire un programme Automate en **Liste d'Instructions** permettant le pilotage du système de commande du moteur tournant dans les deux sens de marche.

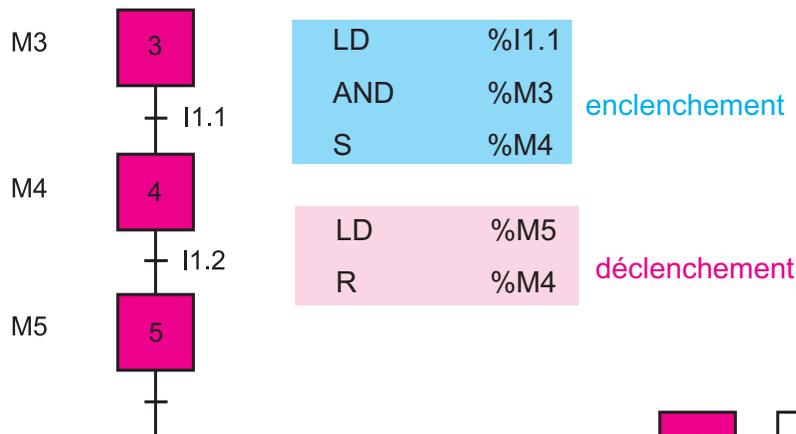
Sachant qu'un programme en liste d'instructions est écrit sous forme de blocs dont chacun commence par l'instruction **LD**, et en tenant compte de la liste des instructions et leur signification :

1- Programmer les modules-étapes :

Chaque module est associé à une variable interne **%Mi**. Chaque variable choisie sera considérée comme une bobine avec une mise à 1 et une remise à 0.

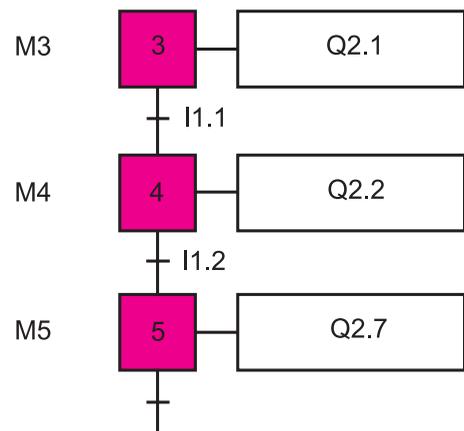
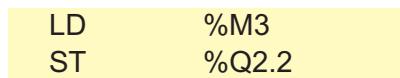
Exemple :

Programmation du module 4



2- Programmer les actions associées :

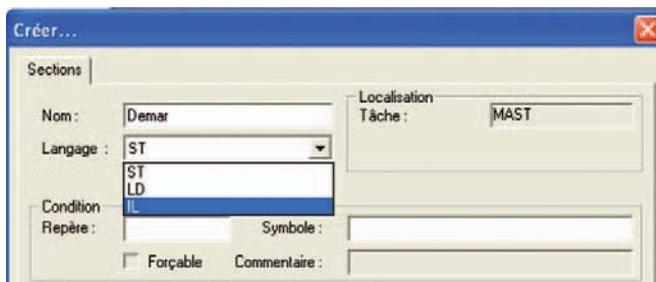
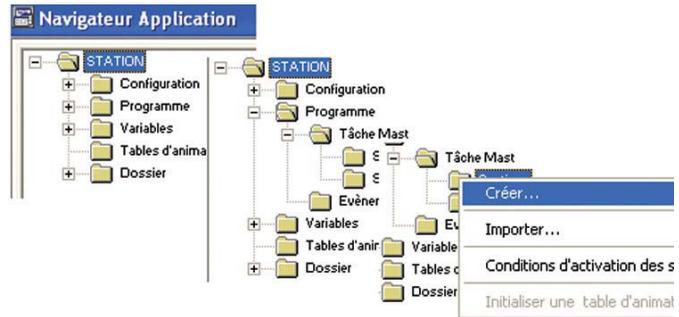
Programmation de l'action associée à l'étape 4



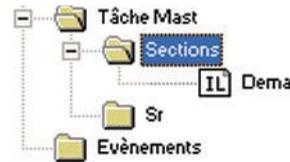
3- Saisie d'un programme LI (liste d'instructions) :

- lancer le logiciel PL7
- menu "Fichier" et "Nouveau"
- "Navigateur d'application".

- cliquer sur "Programme"
- cliquer sur "Tâche Mast"
- cliquer sur "Section" avec le bouton droit
- choisir "Créer"



- saisir le nom de la section
- choisir le langage qui va être utilisé
- valider par OK pour obtenir:



- cliquer sur IL pour démarrer la saisie

4- Insertion d'un temporisateur

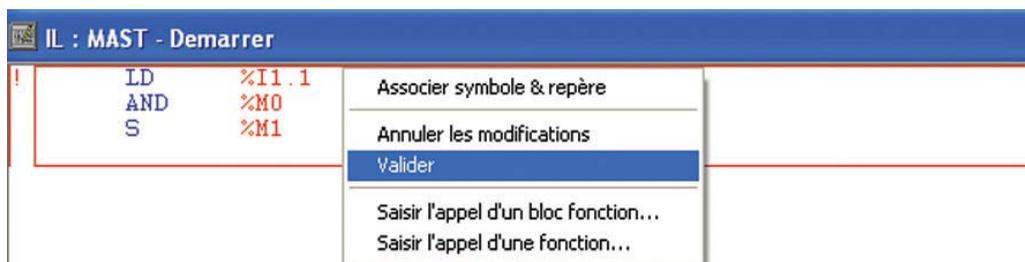
- dans la page de saisie cliquer sur le bouton droit
- choisir 
- choisir 
- on obtient le bloc ci-contre auquel on ajoute :

```
BLK      %TM
IN
OUT_BLK
LD
END_BLK
```

- * avant "IN" la condition d'excitation de la bobine du temporisateur
- * l'ordre du bloc temporisation devant %TM
- * utiliser la sortie Q dans le reste du programme comme %TMI.Q

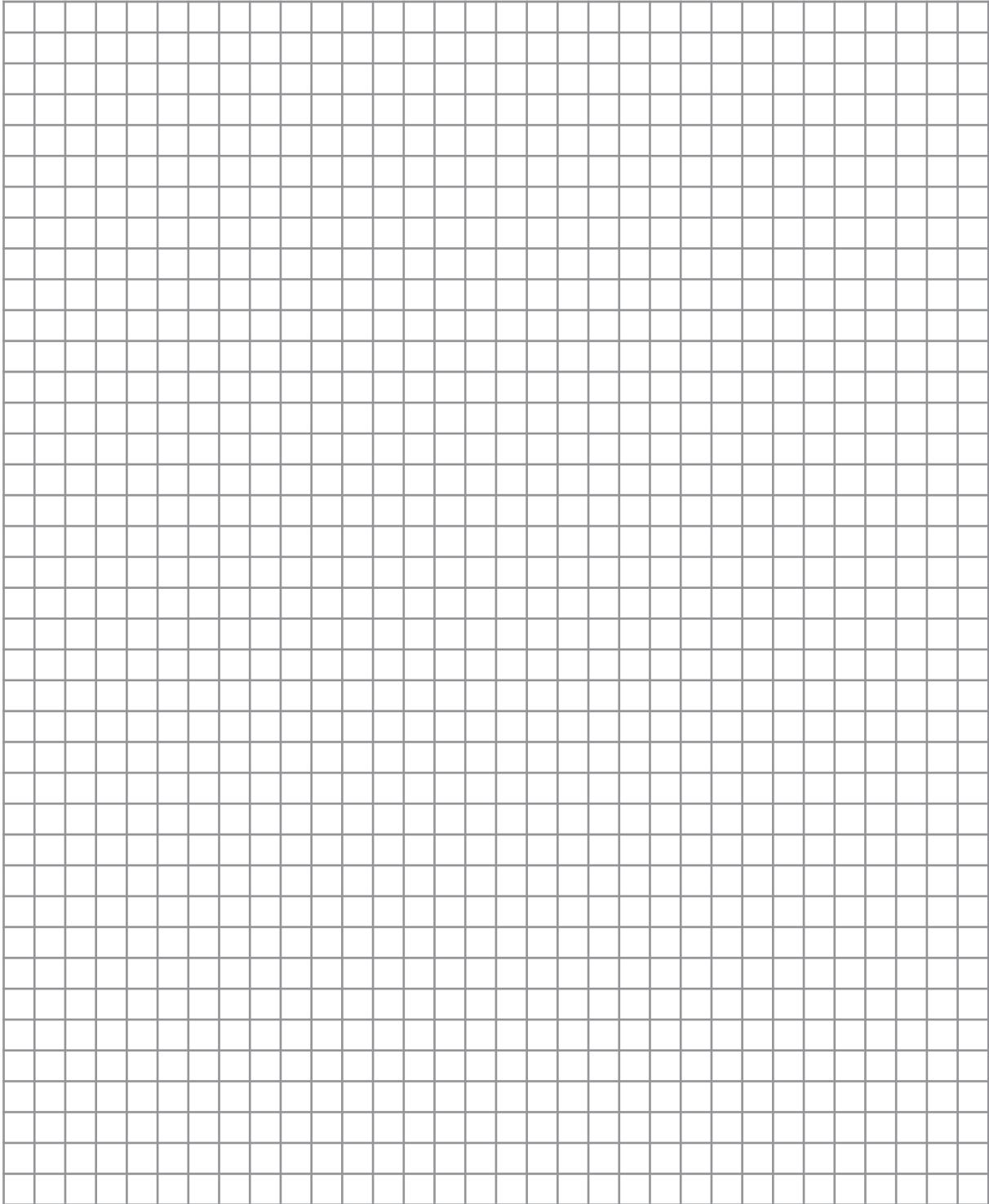
5- Validation du programme saisi

Après avoir terminé, cliquer sur le bouton droit et valider le programme. En cas d'erreur de saisie, la faute est signalée par la couleur rouge.





5- Copier le programme obtenu sur la grille ci-dessous :



6- Charger le programme dans l'Automate et vérifier le fonctionnement :
suivre minutieusement les étapes décrites dans l'activité pratique N°2.

Activité 5

Il s'agit de traduire un GRAFCET codé automate en un programme Langage GRAFCET permettant de piloter un système de commande d'un moteur asynchrone dans les deux sens de marche.

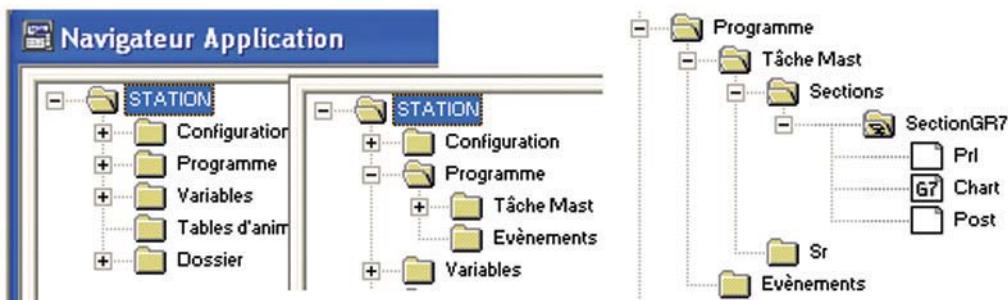
A partir du **GRAFCET codé Automate** utilisé dans l'activité précédente, réécrire un programme Automate en **Langage GRAFCET** permettant le pilotage du système de commande du moteur tournant dans les deux sens de marche.

1- Programmer les modules-étapes :

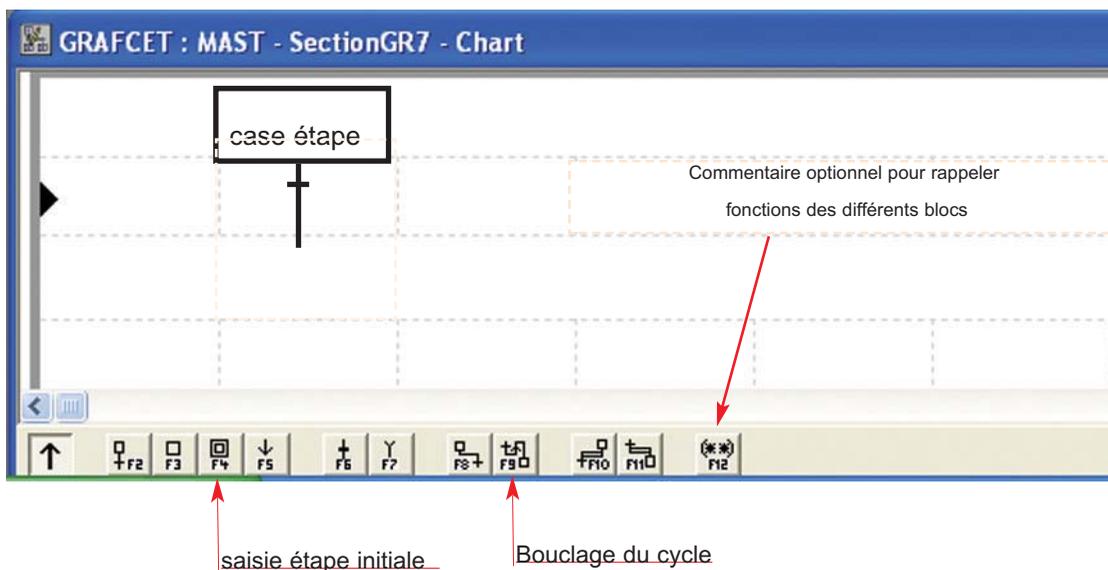
Saisir le GRAFCET à l'écran du PL7 en suivant les instructions suivantes :

-configurer le nouveau fichier en choisissant le type d'Automate et en déclarant l'option GRAFCET.

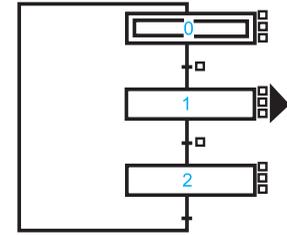
-dans le "Navigateur Application", cliquer successivement sur "Programme" puis "Tâche Mast" puis "Sections" et enfin "G7"



-commencer à tracer le GRAFCET dans la grille proposée par le logiciel PL7 en utilisant les outils qui apparaissent en bas et à gauche de l'écran.

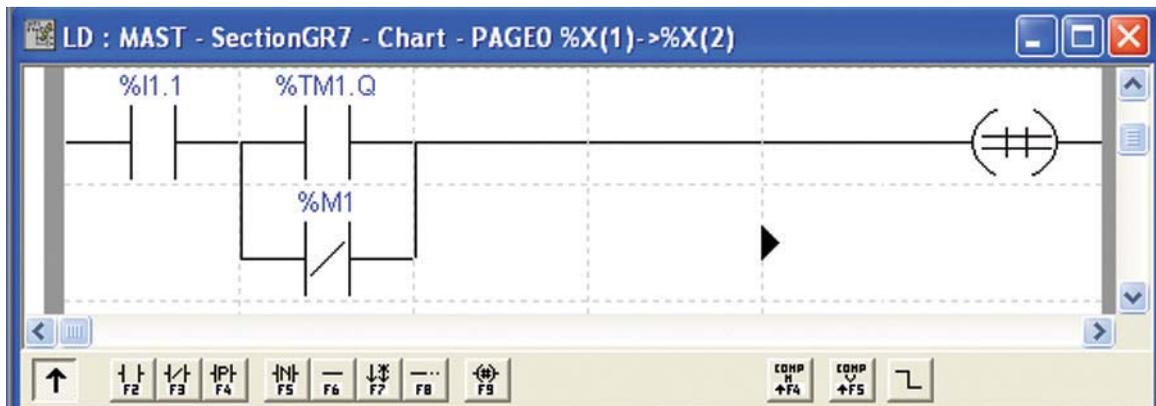
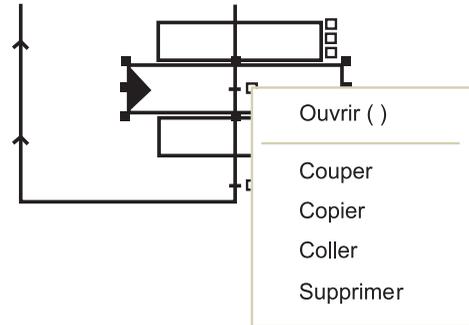


- après avoir terminé le tracé du GRAFCET on procède à la validation par appui sur le bouton droit de la souris annoncée par un changement de la couleur du tracé (du rouge au noir).



2- Programmer les réceptivités :

- pointer la transistion dont on veut programmer la réceptivité et cliquer avec le bouton droit.
- choisir l'option "Ouvrir ()"
- la fenêtre de choix du langage d'expression s'ouvre
- cliquer sur LD pour exprimer la condition en langage LADER (réseaux à contacts)
- rassembler en parallèle et en série différents types de contacts suivant la réceptivité à programmer et suivant les affectations préférées pour exciter une bobine "Dièse". A la fin de la saisie, on valide par appui sur le bouton droit de la souris puis "Valider".



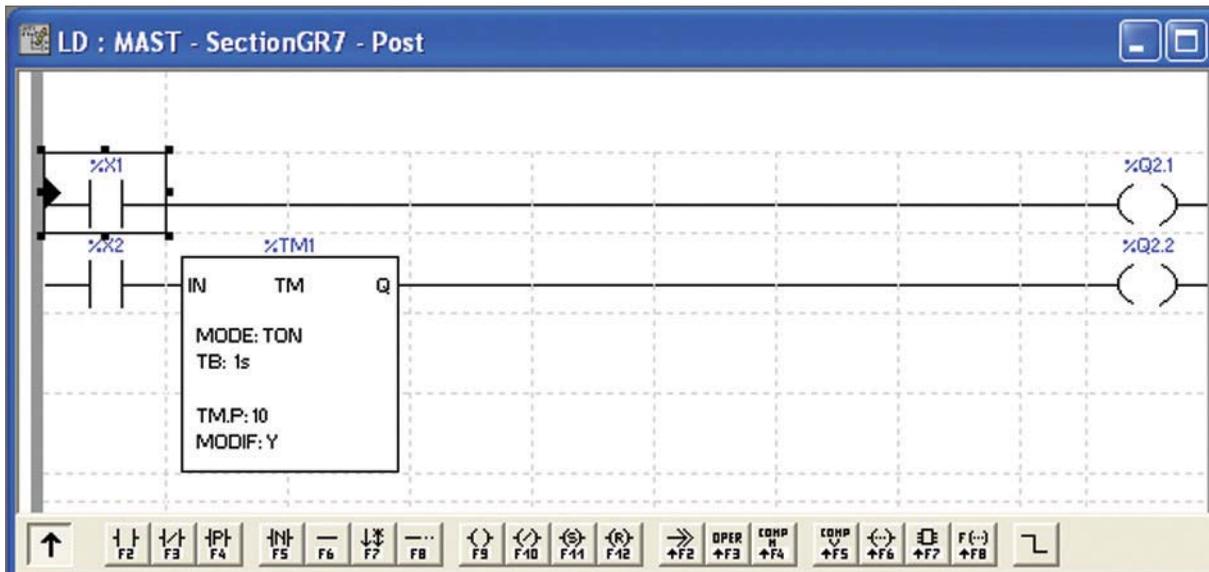
3- Programmer les actions associées :

- revenir au "Navigateur Application"
- cliquer sur "Post"
- éditer les propriétés de "Post" en choisissant le langage Lader (LD)
- tracer le schéma de cablage de chaque action associée aux étapes du GRFCET et ce en tenant compte des affectations préférées et de l'état de la mémoire étape %Xi (i étant le numéro de l'étape).

NB : Les temporisateurs, les temporisateurs series7, les monostables, les compteurs, les registres et les Drums sont saisis à partir du bloc "Fonctions Graphiques" (voir photo-écran ci-contre).



4- Changer les propriétés d'un bloc temporisateur :



Les deux principales caractéristiques d'un temporisateur sont :

- la base de temps
- la durée du retard

Pour mettre au point ces deux composantes on revient au "Navigateur Application". On clique sur "Variables" puis sur l'option "SFB". L'écran de réglage suivant apparaît.

Repère	Type	Symbole	Preset	Mode	TB
%TM0	TM	%TM0	9999	TON	1mn
%TM1	TM	%TM1	10	TON	1s
%TM2	TM	%TM2	9999	TON	1mn
%TM3	TM	%TM3	9999	TON	1mn

5- Charger le programme dans l'Automate TSX 3721 :

Suivre minutieusement les étapes décrites dans l'activité pratique N° 2.

Tester le fonctionnement du système de commande du moteur asynchrone.

Imprimer à partir du logiciel PL7 :

- la section GRAFCET
- la section GR7- Chart
- la section GR7-Post

Activité 6

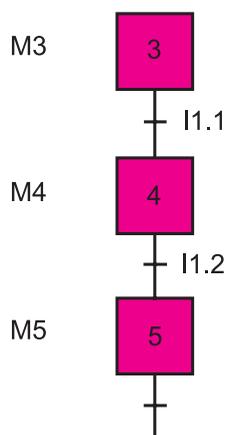
Il s'agit de traduire un GRAFCET codé automate en un programme Liste d'Instructions permettant de piloter un système de commande d'un moteur asynchrone dans les deux sens de marche (**sur Automate AEG**).

A partir du **GRAFCET codé Automate** tracé dans l'activité précédente, écrire un programme Automate en **Liste d'Instructions** permettant le pilotage du système de commande du moteur tournant dans les deux sens de marche.

Sachant qu'un programme en liste d'instructions est écrit sous forme de blocs dont chacun commence par l'instruction **I L**, et en tenant compte de la liste des instructions et leur signification :

1- Programmer les modules-étapes :

Chaque module est associé à une variable interne **Mi**. Chaque variable choisie sera considérée comme une bobine avec une mise à 1 et une remise à 0.

Exemple :**Programmation du module 4**

A	I1
---	----

A	M3
---	----

SL	M4
----	----

enclenchement

A	M5
---	----

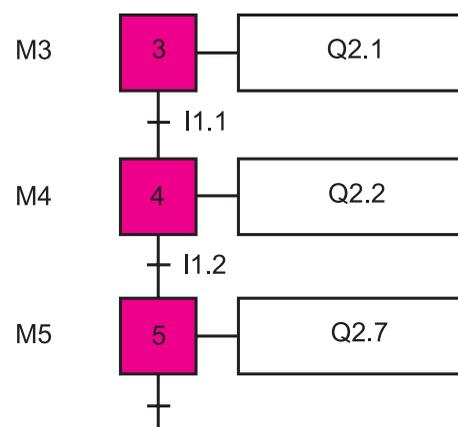
RL	M4
----	----

déclenchement

2- Programmer les actions associées :

programmation de l'action associée à l'étape 4

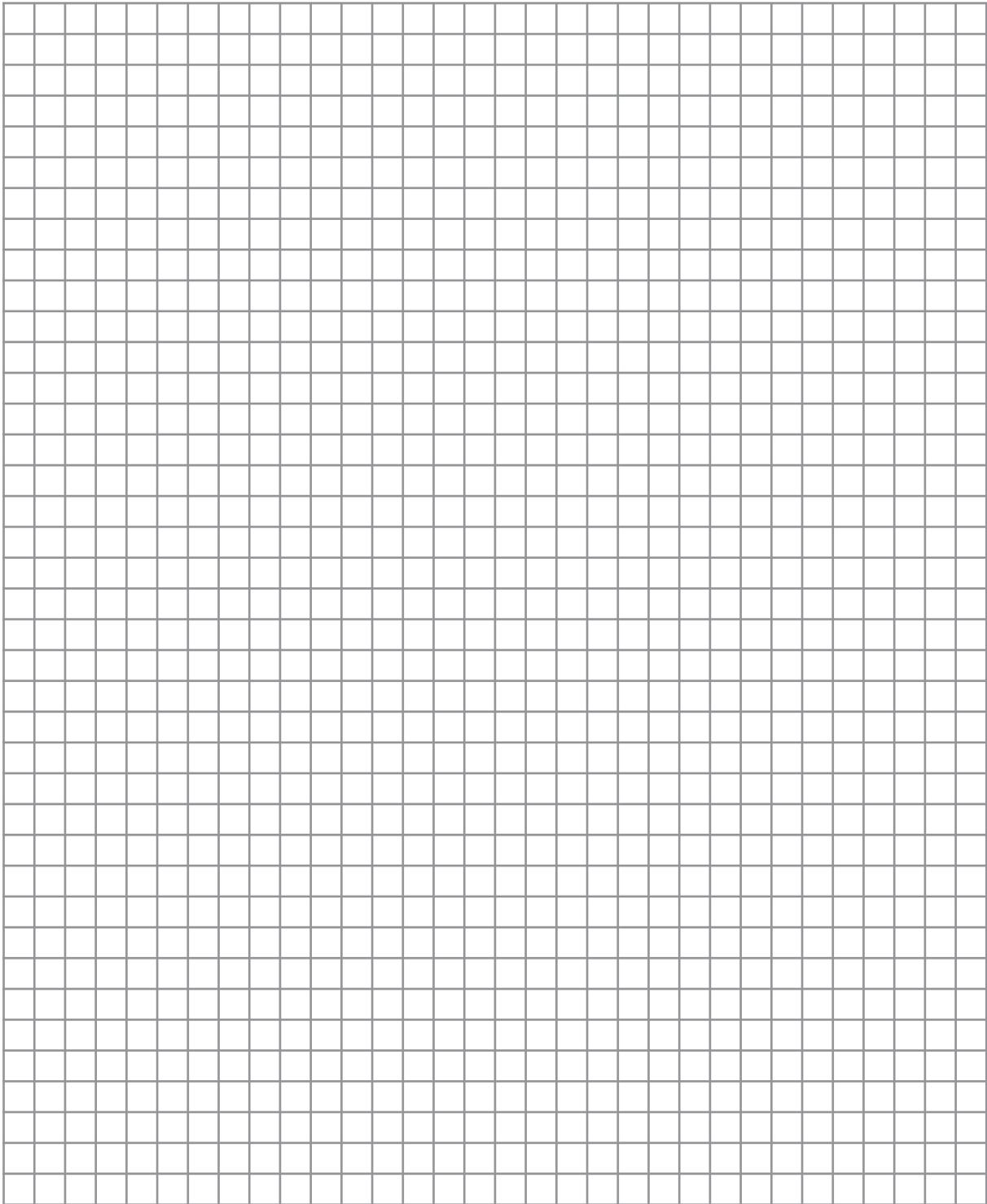
A	M3
=	Q2

**3- Saisie d'un programme IL (liste d'instructions)**

- lancer le logiciel **AKL**
- menu "Fichier" et "Nouveau"
- "Navigateur d'application"



4- Copier le programme obtenu sur la grille ci-dessous :



5- Charger le programme dans l'Automate et vérifier le fonctionnement :

Suivre minutieusement les étapes décrites dans l'activité pratique N° 2.

TP A4-2

Microcontrôleurs

- Objectifs Spécifiques :

OS A44 - Identifier à partir d'une application industrielle et/ou de son dossier technique un microcontrôleur.

OS A45 - Elaborer un programme spécifique à une application à base de microcontrôleur.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

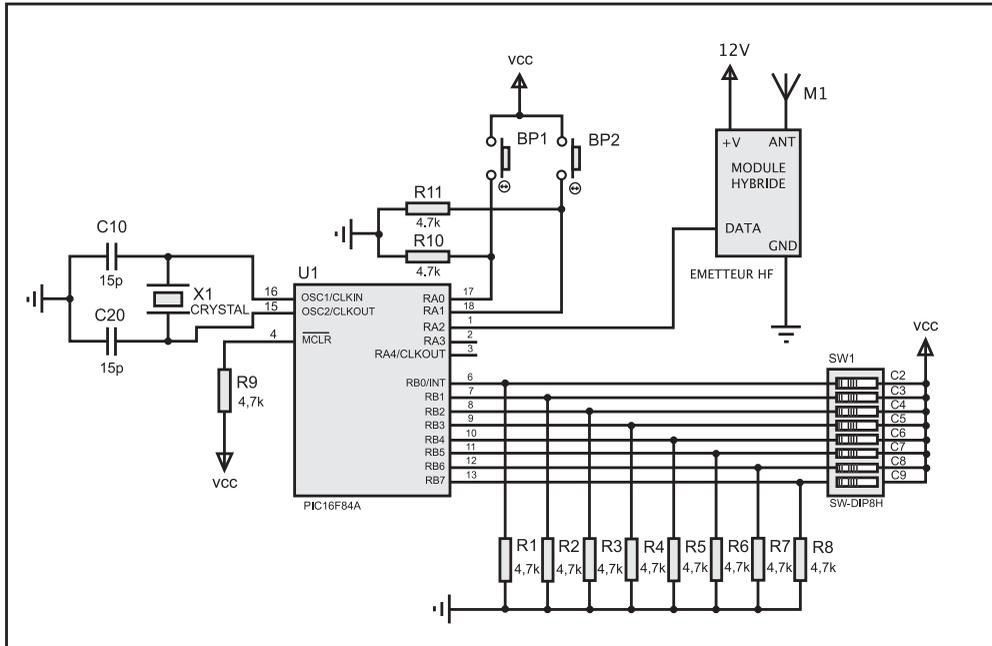
- ✓ Poste PC
- ✓ Logiciel de simulation
- ✓ Matériel d'essai en électronique
- ✓ Maquette
- ✓ Kit de programmation
- ✓ Systèmes techniques à base de microcontrôleur.



Activité 1

Activité de découverte du système télécommande de porte de garage.

On vous rappelle le schéma de la télécommande de la porte automatique du garage.



- 1- Le circuit intégré U1 est un
- 2- Quel est le type de l'horloge
- 3- Donner le rôle de la broche \overline{MCLR} et justifier son état logique.....
- 4- Expliquer le rôle des résistances R10 et R11.....
- 5- Compléter le tableau suivant par : (entrée, sortie, non utilisée)

Port A		Port B	
RA0		RB0	
RA1		RB1	
RA2		RB2	
RA3		RB3	
RA4		RB4	
		RB5	
		RB6	
		RB7	



Activité 2

Il s'agit d'identifier les différentes caractéristiques du microcontrôleur PIC16F84A

- 1- D'après la réponse à la question 5 de l'activité 1 trouver les valeurs q'on devra placer dans les registres TRISA et TRISB du microcontrôleur de la télécommande.

TRISA :

--	--	--	--	--	--	--	--

 = (.....) H

TRISB :

--	--	--	--	--	--	--	--

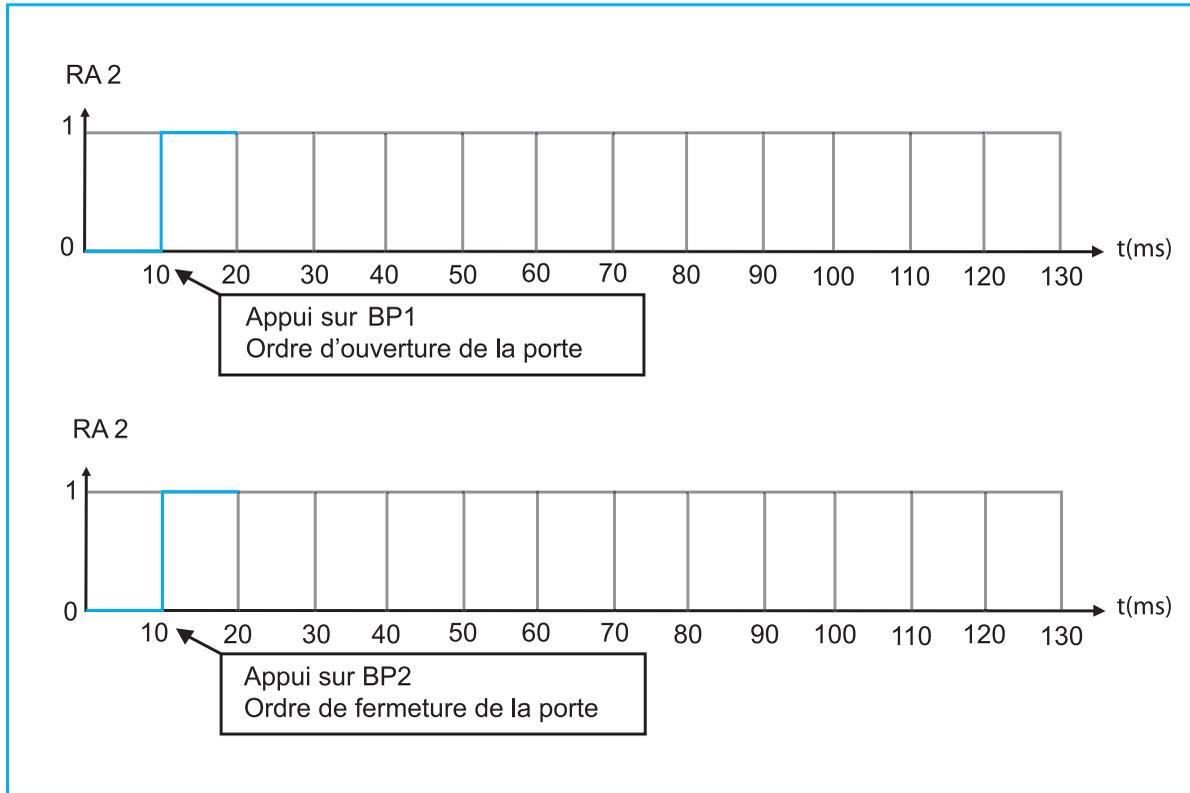
 = (.....) H

- 2- En se référant au document constructeur du 16F84A compléter les phrases suivantes :

- ✓ Le constructeur du PIC16F84 s'appelle :
- ✓ La taille de la mémoire programme est
- ✓ La taille de la mémoire de donnée est
- ✓ La fréquence d'horloge maximale est deMHz
- ✓ Le nombre de broches d'entrée/sortie est
- ✓ L'alimentation du microcontrôleur est deV
- ✓ Il possède une interface parallèle : oui non
- ✓ Il possède une interface série : oui non
- ✓ Il possède un convertisseur CAN : oui non
- ✓ Il possède un chien de garde : oui non



- 3- Compléter les chronogrammes relatifs au signal sur la broche RA2 du microcontrôleur dans les deux cas suivants :
- Emission d'un signal correspondant à un ordre d'ouverture de la porte
 - Emission d'un signal correspondant à un ordre de fermeture de la porte.





Activité 4

Il s'agit de lire correctement un algorithme et en déduire l'algorithme de commande de la télécommande.

- 1- Donner la table de vérité des broches RA0 et RA1 en fonction des broches RB0, RB1 et RB2 du microcontrôleur sachant qu'on a implanté l'algorithme suivant :

Algorithme activité41;

```

_DEBUT
| TRISA←$FC;           // RA0, RA1 : sortie ; les autres broches entrée.
| TRISB←$FF;           // tout le portb est configuré en entrée.
| TANQUE (1=1) FAIRE   // boucle infinie.
|
|   _DEBUT
|   | SI ((Portb.0=1)OU(Portb.1=1)) ALORS Porta.0←1 SINON Porta.0←0; FINSI ;
|   | SI ((Portb.0=1) ET (Portb.2=1)) ALORS Porta.1←1 SINON Porta.1←0; FINSI ;
|   | _FINFAIRE;
|
|_FIN.
    
```

RBO	RB1	RB2	RA0	RA1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- 2- Donner les équations de RA0 et RA1 puis tracer le logigramme correspondant
 RA0 =.....
 RA1=.....

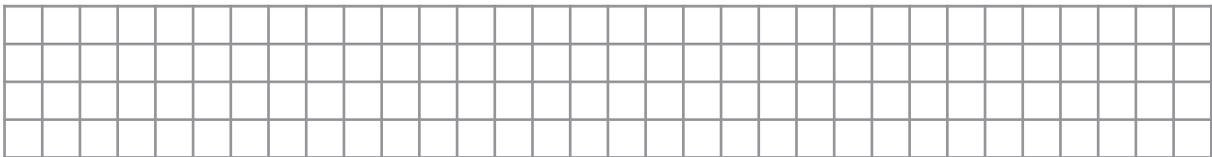


3- Déterminer le rôle de l'algorithme suivant :

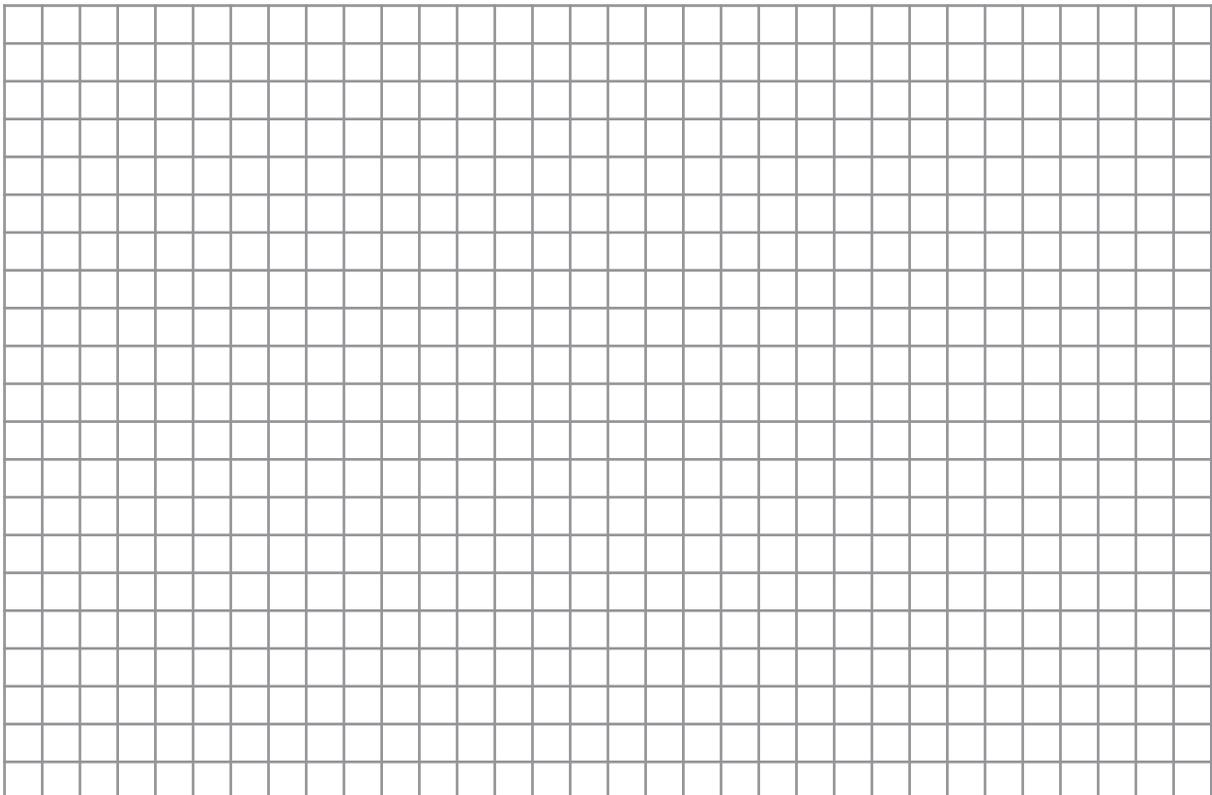
Algorithme activité43;

variables i :entier;

```
_DEBUT
| TRISA←$FB;           // RA0, RA1 : sortie ; les autres broches entrée.
| TRISB←$FF;          // tout le portb est configuré en entrée.
| TANQUE (1=1) FAIRE  // boucle infinie.
|   _DEBUT
|   |   POUR i variant de 0 jusqu'à 7 FAIRE
|   |   |   _DEBUT
|   |   |   |   SI (portB.i=1) ALORS PortA.2←1 SINON porta.2←0; FINSI;
|   |   |   |   Attente (10ms);
|   |   |   |   _FINFAIRE;
|   |   |   _FINFAIRE;
|   _FINFAIRE;
| _FIN.
```



4- Ecrire un algorithme pour la télécommande de la porte automatique de garage.





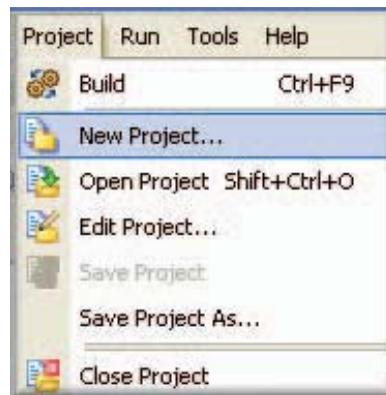
Activité 5

Il s'agit d'écrire un programme en mikropascal pour le microcontrôleur de la télécommande afin de simuler le fonctionnement puis le tester sur plaque d'essai

1- Création d'un projet sur Mikropascal :

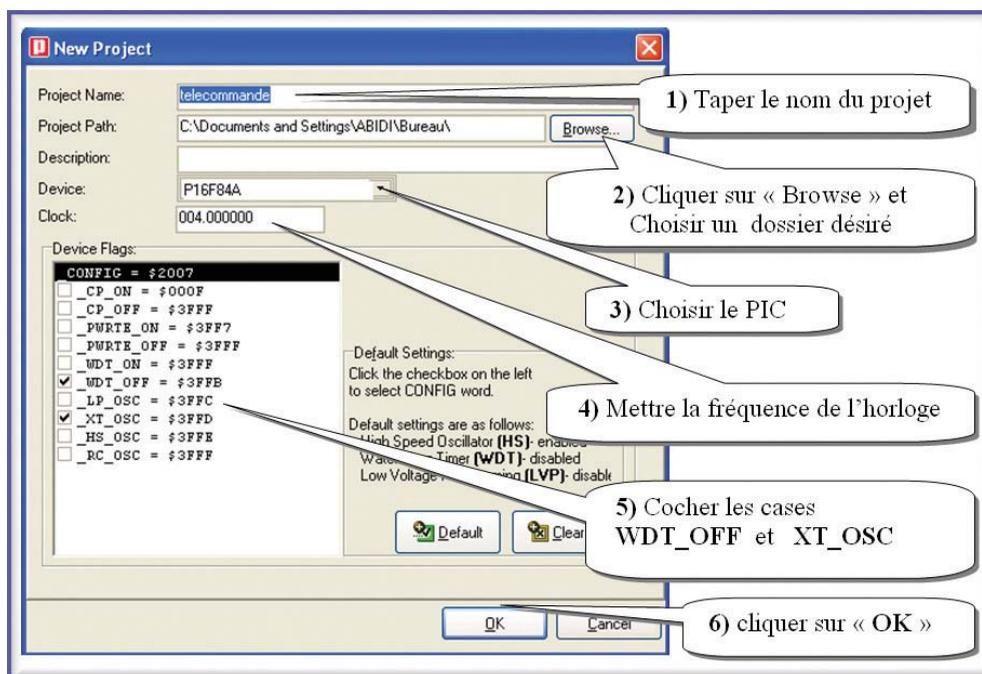
Première étape : Edition du projet

Lancer l'application Mikropascal puis sélectionner le menu " Project 'new Project



Deuxième étape :

Remplir la boîte de dialogue suivante avec les valeurs désirées.



Troisième étape : Saisie du programme

Lors de la fermeture de la fenêtre précédente on peut écrire notre programme sur l'éditeur de code comme suit :



```

1 program telecommande;
2
3 procedure trans_code;
4 var i : byte;
5 begin
6 for i:=0 to 7 do
7     begin

```

Saisir le programme suivant :

```

program Telecommande;
procedure trans_code;
var i : byte;
begin
    for i:=0 to 7 do
        begin
            if PortB.i =1 then setbit(porta,2)
            else clearbit(porta,2); // RA0 reçoit la valeur de RBi
            delay_ms(10); // attente de 10 ms
        end;
    end;
begin
    trisA:=$FB; // RA0, RA1 : entrées ; RA2 : sortie
    trisB:=$FF; // tout le portB : entrées
    while (1=1) // boucle infinie
    do
        begin
            PortA.2 :=0; // initialisation de RA0
            if portA.0=1 then // ordre de fermeture de la porte
                begin
                    PortA.2 :=1;
                    delay_ms(20) ; // attente de 20 ms pour émettre de C0 et C1
                    trans_code; // émission du code d'identification de C2 à C9
                end;
            if portA.1=1 then
                begin
                    PortA.2 :=1; // émission de C0
                    delay_ms(10) ; // attente de 10 ms
                    PortA.2 :=0; // émission de C1
                    delay_ms(10) ; // attente de 10 ms
                    trans_code; // émission du code d'identification de C2 à C9
                end;
        end;
    end.

```



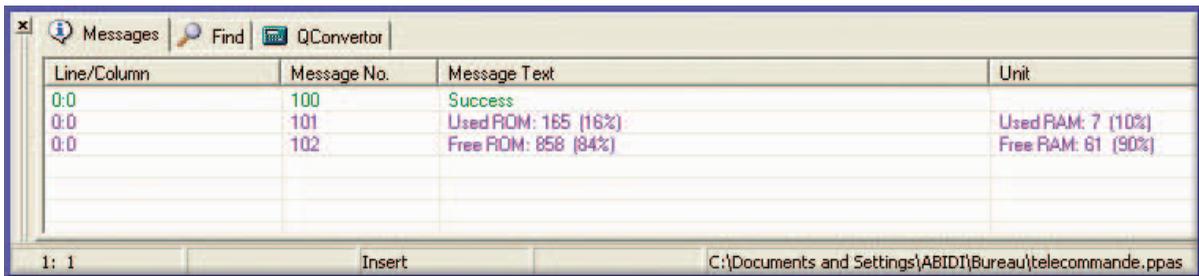
Quatrième étape : Compilation

Après l'edition du programme on clique sur le menu " Project'build "



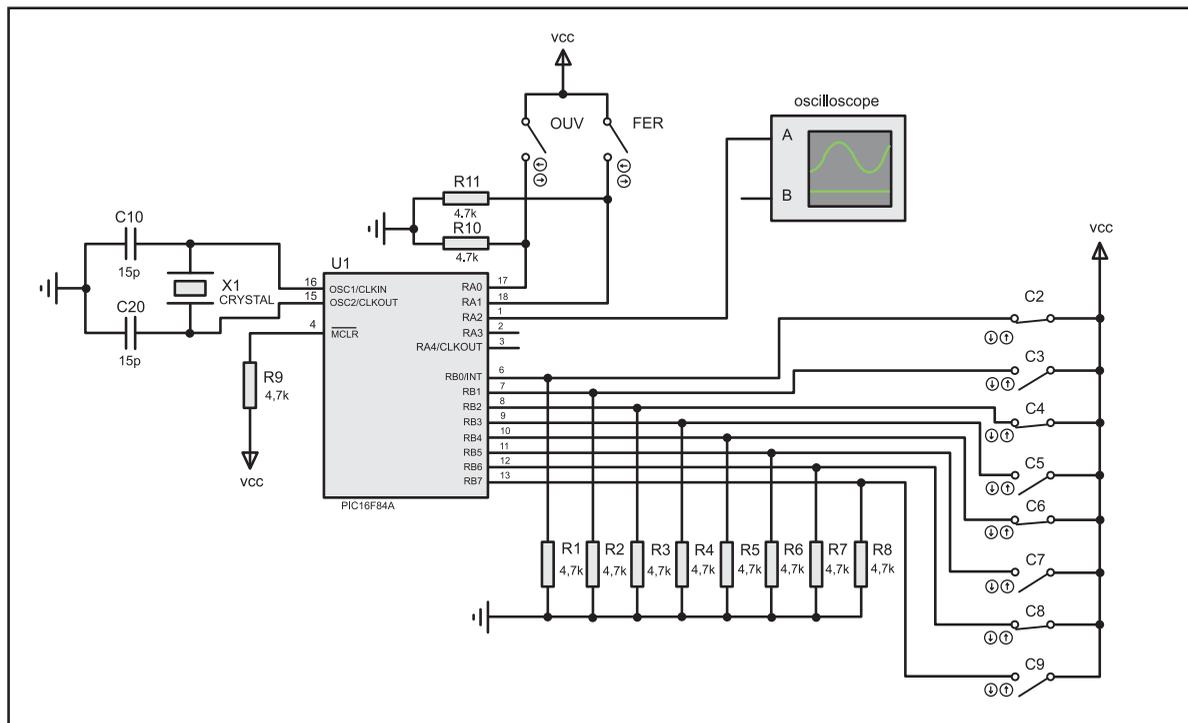
La barre de progression vous informera au sujet du statut de compilation. S'il y a des erreurs, on vous l'annoncera dans la fenêtre d'erreur

Si aucune erreur n'est produite, le message suivant sera affiché et le fichier télécommande.hex sera crée dans le répertoire de travail.



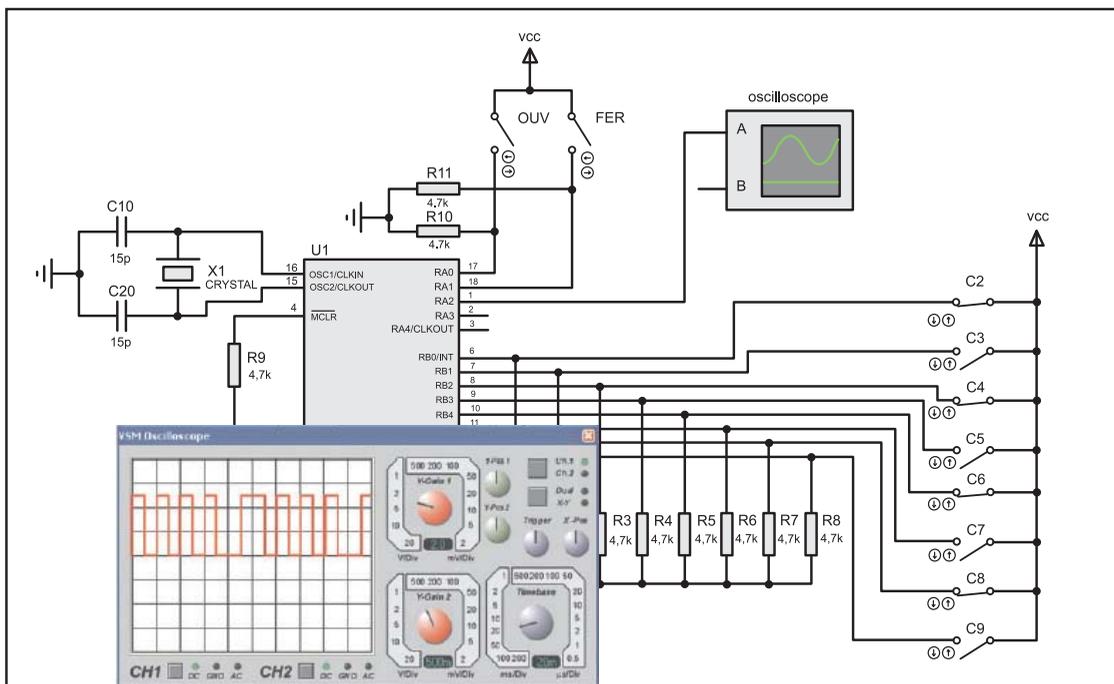
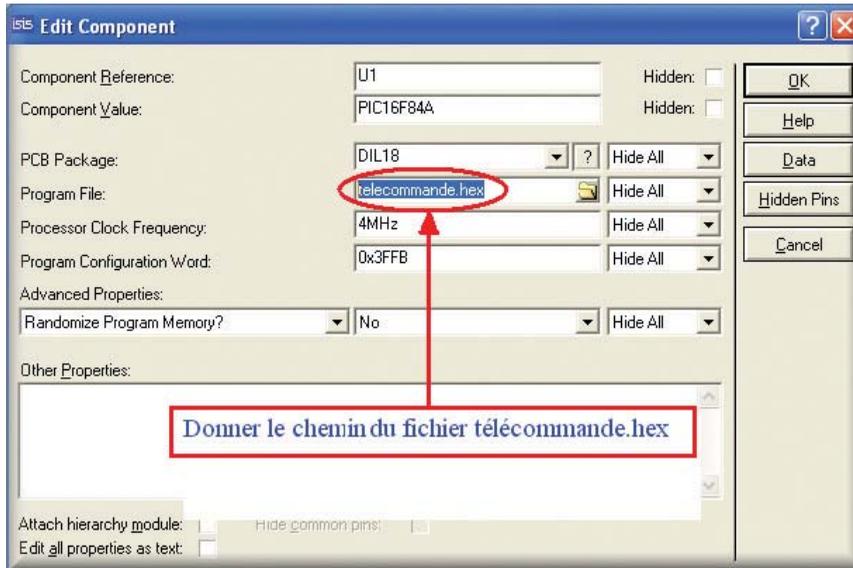
Cinquième étape : Simulation

La simulation du programme peut se faire facilement avec le logiciel ISIS.





Charger le programme dans le microcontrôleur.

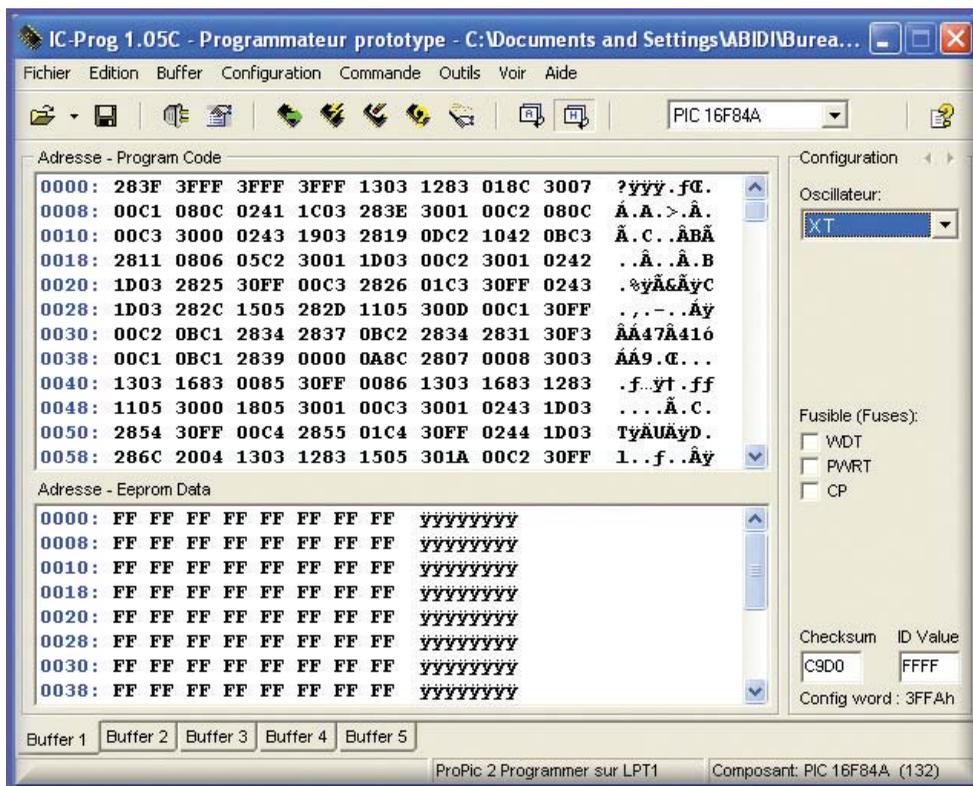


NB : Si vous fermez l'oscilloscope pendant la simulation il n'apparaîtra pas lorsque vous relancer la simulation une deuxième fois, il faut le supprimer et placer un autre oscilloscope.



Sixième étape : Chargement du programme dans le PIC16F84A

Lancer l'application IC-Prog puis sélectionner le type du microcontrôleur et enfin charger le fichier télécommande.hex.



Brancher le programmeur sur le port du PC puis placer le microcontrôleur et lancer le chargement du programme.

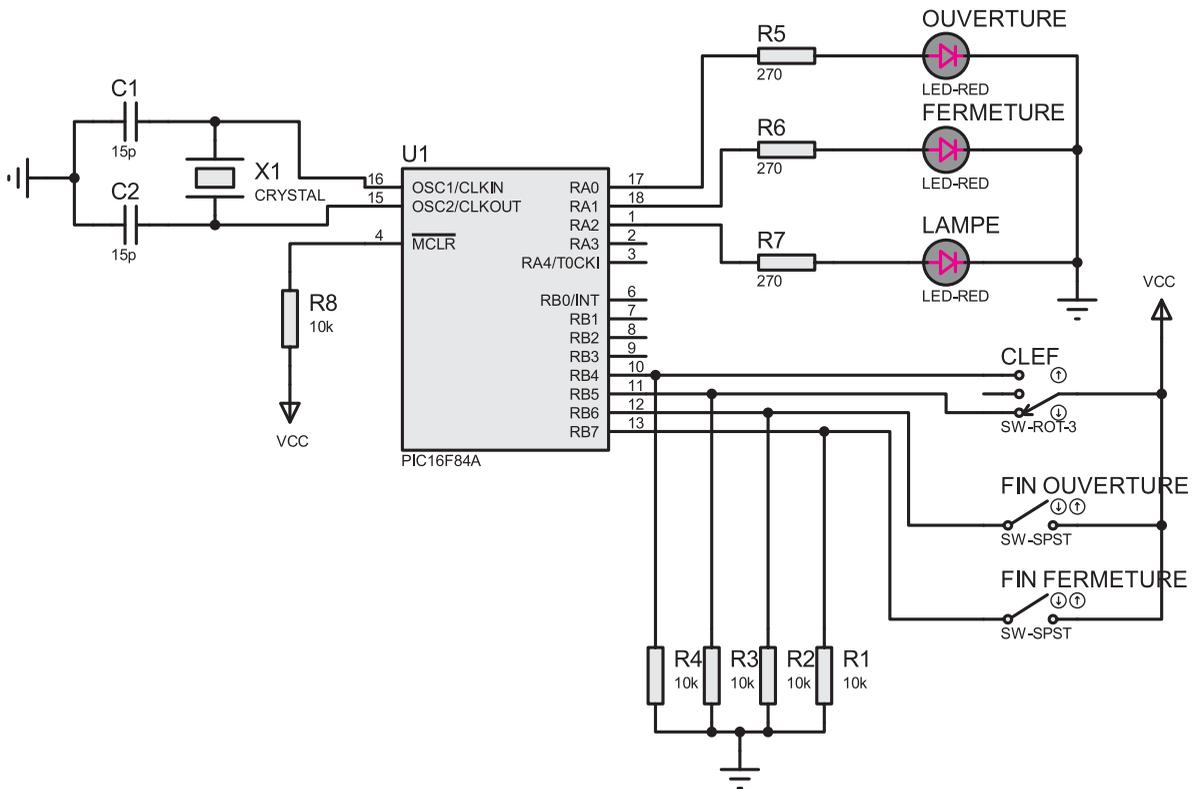




Activité 6

Il s'agit d'écrire un programme en mikropascal afin de simuler le fonctionnement de la carte de commande de la porte du garage.

- 1- En vue de simuler le fonctionnement de la carte de commande de la porte du garage sur le logiciel ISIS, on réalise le schéma suivant :





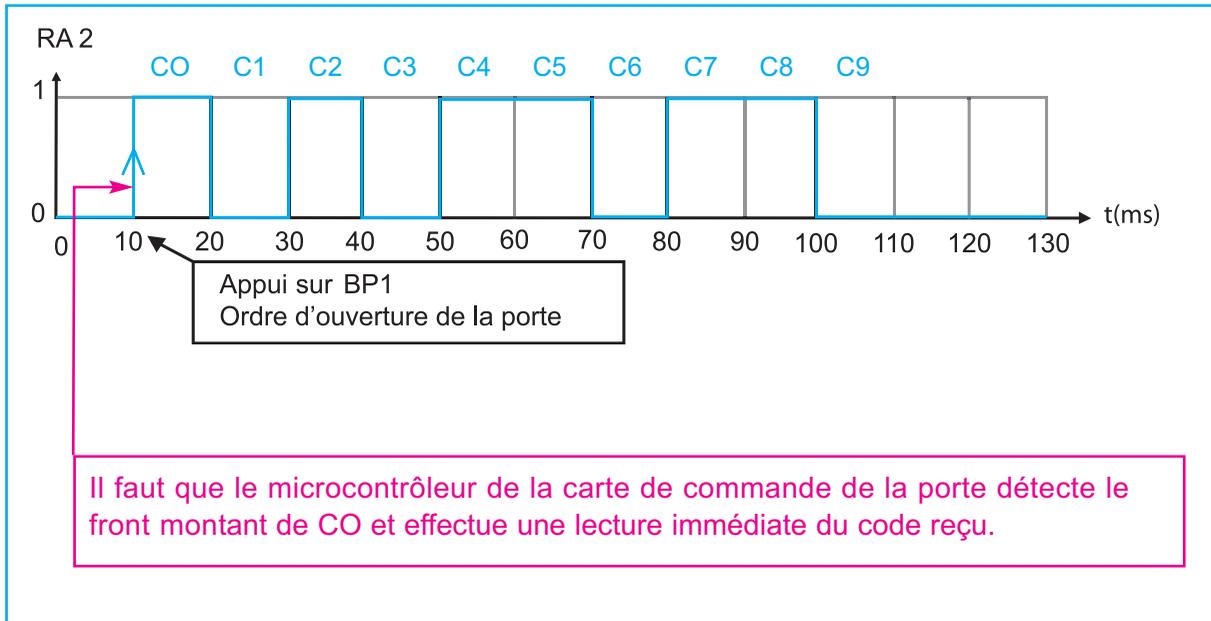
On vous donne l'algorithme relatif au fonctionnement de la carte de commande :
 - Etablir le programme correspondant puis le simuler.

```

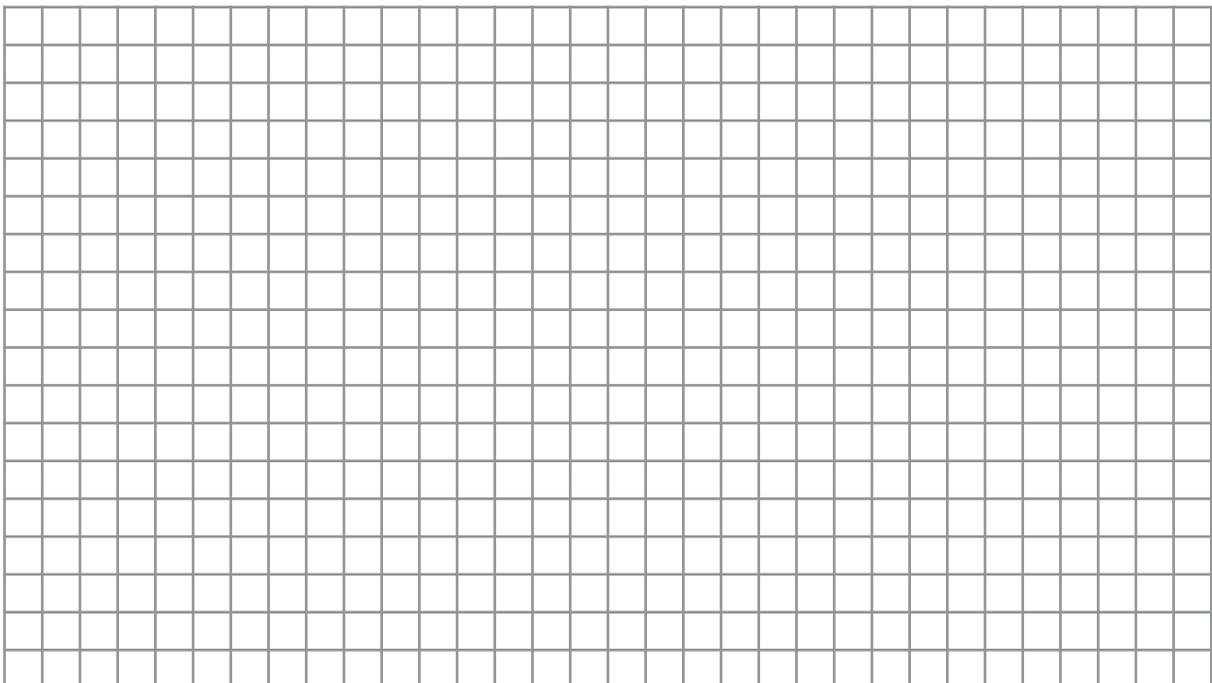
Algorithme Carte_commande;
_DEBUT
| TRISB ← $F8;           // RA0 , RA1, RA2 : sorties
| TRISB ← $FF ;        // tout le port b est configuré en entrée
| porta ← 0;           // initialisation des sorties
| TANQUE (1=1) FAIRE    // boucle infinie
|   _DEBUT
|   | SI (portb.4=1) ALORS // commutateur en position ouverture de la porte
|   |   _DEBUT
|   |   | TANQUE ((portB.6=0) ET (portb.4=1))
|   |   |   FAIRE // tant que ordre d'ouverture de la porte
|   |   |   // et capteur de fin de course non actionné
|   |   |   _DEBUT
|   |   |   | porta.0←1; // actionner moteur en rotation en sens d'ouverture
|   |   |   | porta.2←1; // allumé la lampe de signalisation
|   |   |   | _FINFAIRE ;
|   |   | _FINSI ;
|   | SI (portb.5=1) ALORS // commutateur en position fermeture de la porte
|   |   _DEBUT
|   |   | TANQUE ((portB.7=0) ET (portb.5=1))
|   |   |   FAIRE // tant que ordre de fermeture de la porte
|   |   |   //et capteur de fin de course non actionné
|   |   |   _DEBUT
|   |   |   | porta.1←1; // actionner moteur en rotation en sens de fermeture
|   |   |   | porta.2←1; // allumé la lampe de signalisation
|   |   |   | _FINFAIRE;
|   |   | _FINSI ;
|   |   porta←0; // arrêter le moteur et éteindre la lampe
|   | _FINFAIRE;
| _FIN.
    
```




Pour assurer une bonne réception du code transmit par la télécommande il faut que la carte de commande de la porte détecte la transmission dès que le code C0 soit émis. C'est-à-dire qu'il faut assurer une synchronisation entre les deux systèmes.



Alors comment peut on introduire dans ce programme une procédure qui répond au code de la télécommande ?





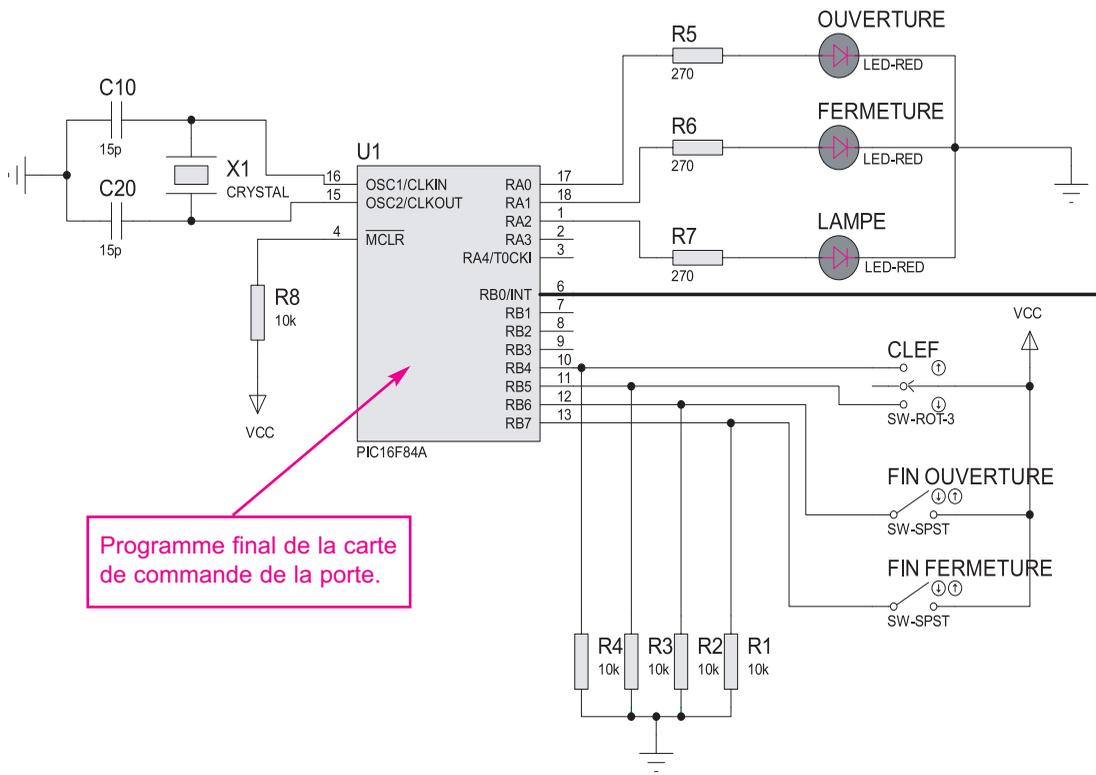
```

while ((portB.6=0) and (portb.4=1)) do // tant que ordre d'ouverture de la porte
                                        // et capteur de fin de course non actionné
    begin
        porta.0:=1; // actionner moteur en rotation en sens d'ouverture
        porta.2:=1; // allumé la lampe de signalisation
    end;
if (portb.5=1) then // commutateur en position fermeture de la porte
    while ((portB.7=0) and (portb.5=1)) do // tan que ordre de fermeture de la porte
                                        //et capteur de fin de course non actionné
        begin
            porta.1:=1; // actionner moteur en rotation en sens de fermeture
            porta.2:=1; // allumé la lampe de signalisation
        end;
if (code=%0110110101) then // code correspondant à un ordre d'ouverture
    while (portB.6=0) do // tan que capteur de fin de course non actionné
        begin
            porta.0:=1; //actionner moteur en rotation en sens d'ouverture
            porta.2:=1; // allumé la lampe de signalisation
            code:=0; // remise à zéro du code
        end;
if (code=%0110110111) then // code correspondant à un ordre de fermeture
    while (portB.7=0) do // tan que capteur de fin de course non actionné
        begin
            porta.1:=1; //actionner moteur en rotation en sens de fermeture
            porta.2:=1; // allumé la lampe de signalisation
            code:=0; // remise à zéro du code
        end;
    porta:=0; // remise à zéro des sorties.
end;
end.

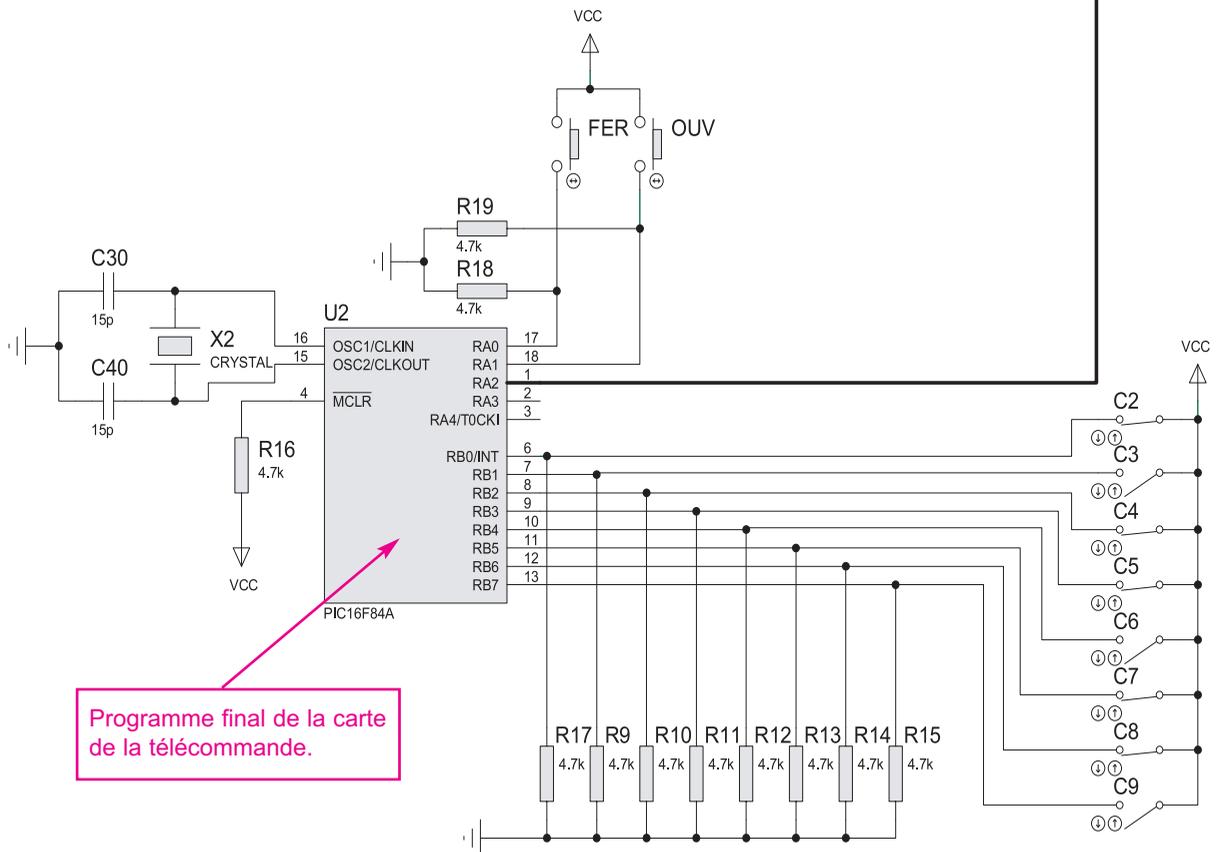
```

3- Compiler le programme et simuler le fonctionnement par le logiciel ISIS.

NB : relier la broche RA2 du microcontrôleur de la télécommande à la broche RB0 du microcontrôleur de la carte de commande de la porte pour simuler la transmission du code.



Programme final de la carte de commande de la porte.



Programme final de la carte de la télécommande.



Les librairies les plus utilisées en Mikropascal

- LCD Library
- Keypad Library
- Delays Library
- Util Library

1- LCD library :

Le compilateur MikroPascal fournit une bibliothèque pour communiquer avec l'afficheur LCD utilisé généralement en mode 4-bit

a) Lcd_Config :

Syntaxe

`Lcd_Config (var port : byte; const. RS, EN, WR, D7, D6, D5, D4 : byte);`

Description

Initialiser l'afficheur LCD et définir les broches du microcontrôleur à relier à l'LCD.

Exemple

```
Lcd_Config (PORTD,1,2,0,3,5,4,6);
```

b) Lcd_Init :

Syntaxe

`Lcd_Init (var port : byte);`

Description

Initialiser l'afficheur LCD avec le PIC en spécifiant le port uniquement, le branchement de l'LCD avec le microcontrôleur est imposé par Mikropascal (consulter l'aide du logiciel)

Exemple

```
Lcd_Init (PORTB);
```

c) Lcd_Out :

Syntaxe

`Lcd_Out (row, col : byte; var text : array[255] of char);`

Description

Ecrire un Texte sur l'afficheur LCD en indiquant sa position (ligne et colonne).

Exemple

Ecrire "Hello!" sur LCD sur la ligne 1, colonne 3:

```
Lcd_Out(1, 3, 'Hello!');
```

d) Lcd_Out_Cp :

Syntaxe

`Lcd_Out_Cp(var text : array[255] of char);`

Description

Ecrire le texte sur l'afficheur LCD à la position actuelle de curseur.

Exemple

Afficher " salut " à la position actuelle de curseur :

```
Lcd_Out_Cp('salut');
```



e) Lcd_Chr :

Syntaxe

`Lcd_Chr(row, col, character : byte);`

Description

Ecrire un Caractère sur l'LCD en indiquant sa position (ligne et colonne).

Exemple

Ecrire "i" sur LCD sur la ligne 2, colonne 3 :

`Lcd_Chr(2, 3, 'i');`

f) Lcd_Chr_Cp :

Syntaxe

`Lcd_Chr_Cp(character : byte);`

Description

Ecrire un caractère sur l'afficheur LCD à la position actuelle de curseur.

Exemple

Ecrire "e" à la position actuelle du curseur.

`Lcd_Chr_Cp('e');`

g) Lcd_Cmd :

Syntaxe

`Lcd_Cmd (command : byte);`

Description

Envoie une commande à l'afficheur LCD. La liste complète des commandes est la suivante :

LCD_FIRST_ROW	Déplacer le curseur à la 1ère ligne
LCD_SECOND_ROW	Déplacer le curseur à la 2 ème ligne
LCD_THIRD_ROW	Déplacer le curseur à la 3 ème ligne
LCD_FOURTH_ROW	Déplacer le curseur à la 4 ème ligne
LCD_CLEAR	Effacer le contenu de l'afficheur LCD
LCD_RETURN_HOME	Retour du Curseur à la position initiale
LCD_CURSOR_OFF	Arrêter le curseur
LCD_MOVE_CURSOR_LEFT	Déplacer le curseur à gauche
LCD_MOVE_CURSOR_RIGHT	Déplacer le curseur à droite
LCD_TURN_ON	Activer l'affichage sur l'afficheur LCD



LCD_TURN_OFF	Arrêter l'affichage sur l'afficheur LCD
LCD_SHIFT_LEFT	Décalage de l'affichage à gauche
LCD_SHIFT_RIGHT	Décalage de l'affichage à droite

2- Keypad Library :

Le MikroPascal fournit une bibliothèque pour travailler avec les claviers matriciels.

a) Keypad_Init :

Syntaxe

`Keypad_Init(var port : word);`

Description

Initialiser et préciser le port sur le quel est branché le clavier.

Exemple

```
Keypad_Init(PORTB);
```

b) Keypad_Read :

Syntaxe

`Keypad_Read : byte;`

Description

Vérifier si une touche est appuyée. La fonction renvoie 1 à 15, selon la touche appuyée, ou 0 si aucune touche n'est actionnée.

Exemple

```
kp := Keypad_Read;
```

c) Keypad_Released :

Syntaxe

`Keypad_Released : byte;`

Description

L'appel de Keypad_Released génère une attente jusqu'à ce qu'une touche soit appuyée et libérée. Une fois libérée, la fonction renvoie 1 à 15, selon la touche.

Exemples

```
kp := Keypad_Released;
```

3- Delays Library :

MikroPascal fournit une procédure pour créer des temporisations.

a) Delay_us :

Syntaxe

`Delay_us(const time_in_us : word);`

Description

Crée un retard dont la durée en microsecondes est time_in_us (une constante). La gamme des constantes applicables dépend de la fréquence de l'oscillateur

Exemple

```
Delay_us(10); // Dix microsecondes
```



Crée un retard dont la durée en millisecondes est `time_in_ms` (une constante). La gamme des constantes applicables dépend de la fréquence de l'oscillateur.

Exemple

```
Delay_ms(1000); // une seconde.
```

c) **Vdelay_ms** :

Syntaxe

```
Vdelay_ms(time_in_ms : word);
```

Description

Crée un retard dont la durée en millisecondes est `time_in_ms` (une variable).

Exemple

```
Var Pause : integer ;
```

```
//...
```

```
Pause := 1000;
```

```
// ...
```

```
Vdelay_ms(pause); // pause d'une seconde
```

d) **Delay_Cyc** :

Syntaxe

```
Delay_Cyc(cycles_div_by_10 : byte);
```

Description

Crée un retard dont la durée est fonction de l'horloge du microcontrôleur.

Exemple

```
Delay_Cyc(10);
```

4- Util Library

La bibliothèque 'util library' contient des diverses routines utiles pour le développement de programmes.

Button

Syntaxe

```
Button (var port : byte; pin, time, active_state : byte) : byte;
```

Description

Les Paramètres (port) et (pin) indiquent l'endroit du bouton ; le paramètre temps d'appui nécessaire pour rendre le bouton actif ; le paramètre `active_state`, peut être 0 ou 1, il détermine si le bouton est ouvert au repos ou fermé au repos.

Exemple

```
while true do
```

```
begin
```

```
  if Button(PORTB, 0, 1, 1) then oldstate := 255;
```

```
    if oldstate and Button(PORTB, 0, 1, 0) then
```

```
      begin
```

```
        PORTD := not(PORTD);
```

```
        oldstate := 0;
```

```
      end;
```

```
end;
```



5- Autres fonctions intégrées :

Le compilateur mikroPascal fournit un ensemble de fonctions de service intégrées utiles. Les fonctions intégrées n'ont aucune condition spéciale ; vous pouvez les employer dans n'importe quelle partie de votre programme.

a) SetBit :

Syntaxe

`SetBit(var register : byte; bit : byte);`

Description Le Paramètre bit doit être une variable

Exemple

```
SetBit(PORTB, 2); // RB2 =1 ;
```

b) ClearBit :

Syntaxe

`ClearBit(var register : byte; bit : byte);`

Description

Le Paramètre bit doit être variable.

Exemple

```
ClearBit(PORTC, 7); // RC7 =0 ;
```

c) TestBit :

Syntaxe

`TestBit(var register : byte; bit : byte) : byte;`

Description

Cette Fonction teste si le paramètre bit est égal à 1. Si oui, la fonction renvoie 1, si non elle renvoie 0. Le Paramètre bit doit être variable.

Exemple

```
flag := TestBit(PORTB, 2); // flag =1 si RB2 =1
```

Remarque :

Pour plus d'informations et pour télécharger les différents programmes et les schémas de simulation, consulter le site suivant :

www.technologuepro.com

TP A5

Asservissements Linéaires

- Objectifs Spécifiques :

OS A5-1 - Décrire le fonctionnement d'un système asservi

OS A5-2 - Identifier les fonctions de rétroaction

OS A5-3 - Vérifier en vue d'améliorer les performances d'un système asservi.

- Conditions de réalisation :

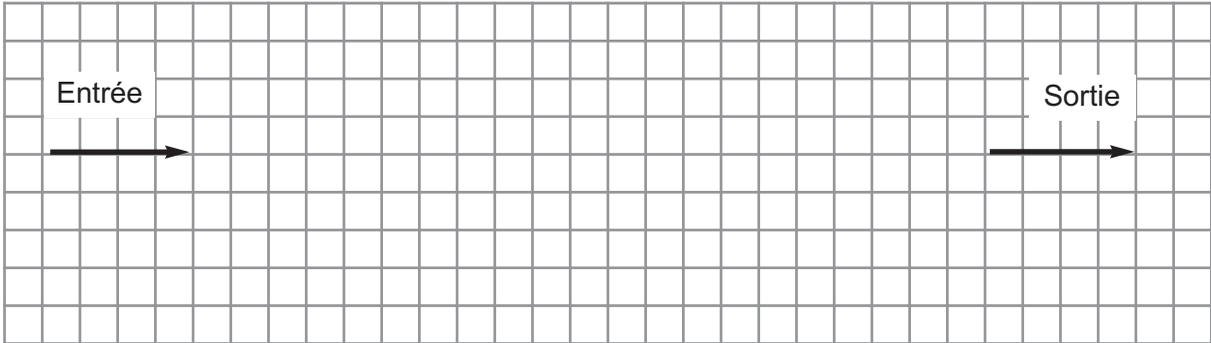
Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

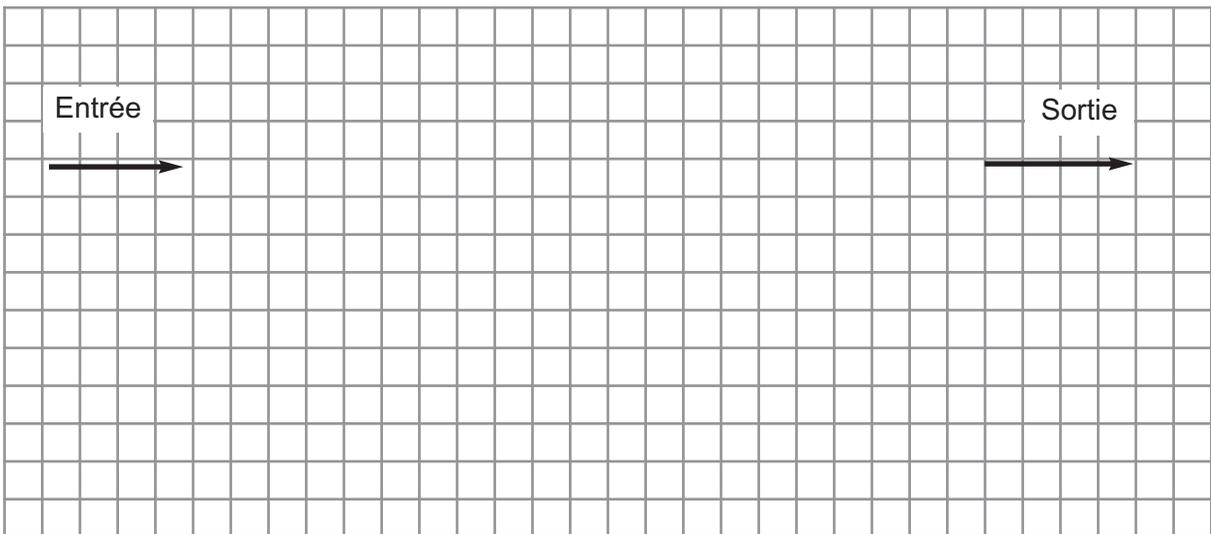
- ✓ Poste PC
- ✓ Logiciel de simulation
- ✓ Maquettes d'asservissement de position et de vitesse
- ✓ Appareils de mesure.



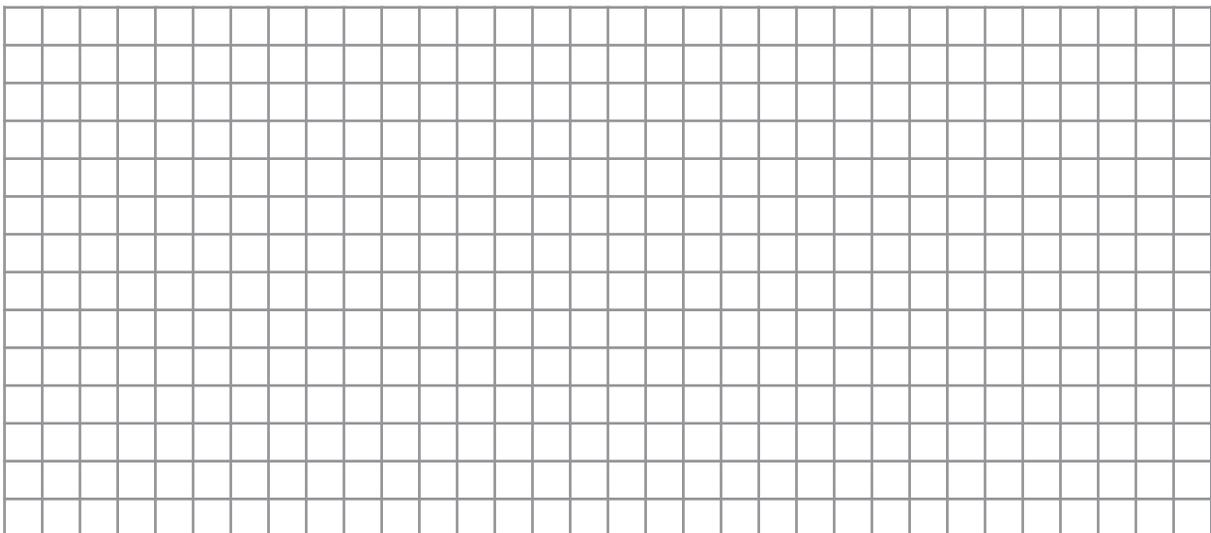
3- Etablir le schéma fonctionnel de ce système fonctionnant en boucle ouverte.



4- Etablir le schéma fonctionnel de ce système fonctionnant en boucle fermée.



5- Comparer les deux modes de fonctionnement.





Activité 2

Il s'agit d'utiliser le logiciel " Correcteur PID " pour mettre en évidence le comportement d'une boucle d'asservissement (brancher et débrancher la boucle). Modifier les actions du correcteur et analyser les performances vis à vis d'une perturbation.

Utiliser le logiciel de simulation "WPID" pour simuler le comportement d'un asservissement de position.



Le logiciel PID pour Windows permet de simuler un système de régulation de position.

Ce logiciel servirait dans le cadre des cours où on enseigne les principes de base des asservissements de procédés. Il permet d'expérimenter la réponse d'un processus simulé en boucle ouverte (BO) ou en boucle fermée (BF).

On peut aussi changer le type de compensation (P, PI, PID ou aucune) .

PID permet de mettre en évidence ce qui se produit quand on ouvre ou on ferme la boucle de mesure sur un système en fonctionnement. Il permet d'observer ce qui se produit quand on ouvre la boucle de mesure d'un système avec un régulateur P, PI ou PID.

Le labo virtuel : L'asservissement PID

SIMULATEUR CACHER **BONUS** A PROPOS QUITTER

Figure 1

Figure 2

Schéma électrique élémentaire

- La position d'un outil sur une machine à commande numérique est asservie à une consigne donnée par le programme d'usinage.

Il faut savoir que les grandeurs physiques se prêtent plus ou moins bien à un asservissement PID. La notion d'inertie est très importante car c'est elle qui va occasionner un dépassement de la consigne dû à "l'élan" du dispositif en raison de son inertie.

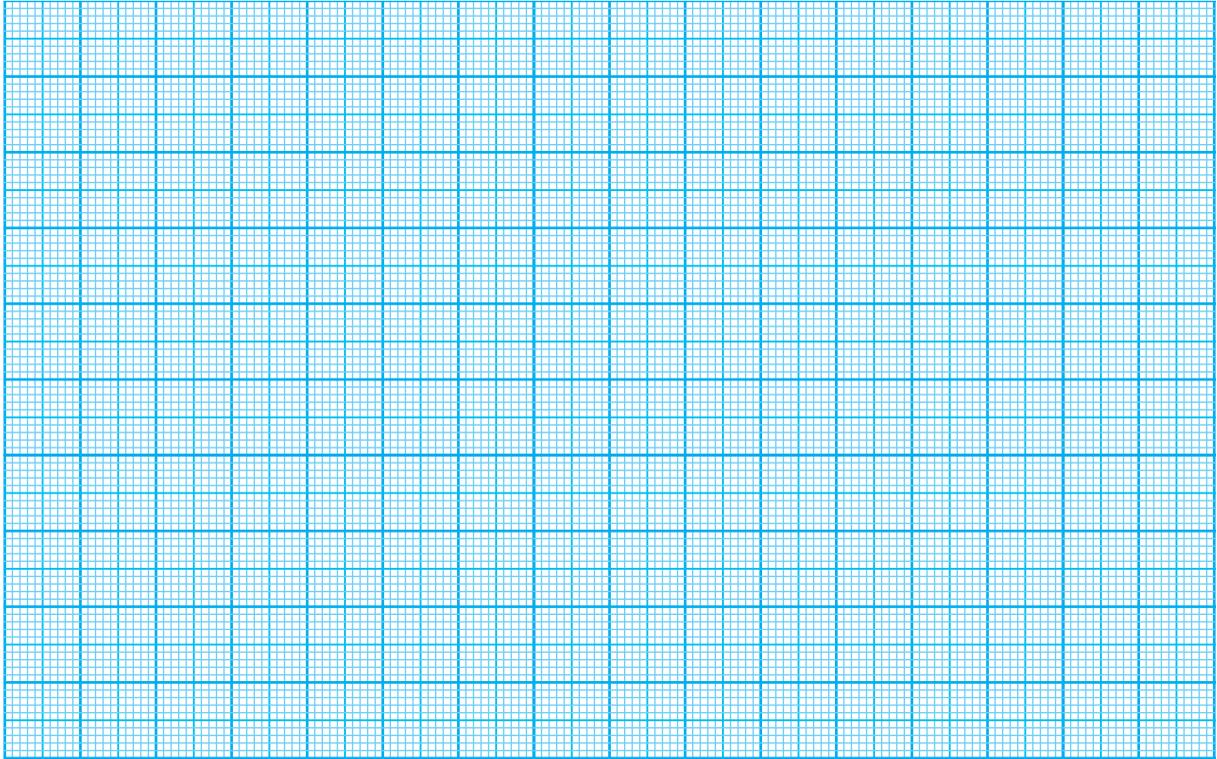
J'ai donc choisi pour cette première approche pédagogique et pour le simulateur, un asservissement de position qui est en fait un exemple très typique.

Il s'agit donc d'asservir la position d'un objet massif à une consigne en agissant automatiquement sur le couple d'un moteur transformé mécaniquement en force linéaire chargé de mouvoir l'objet massif. Il peut s'agir aussi directement d'un moteur linéaire.

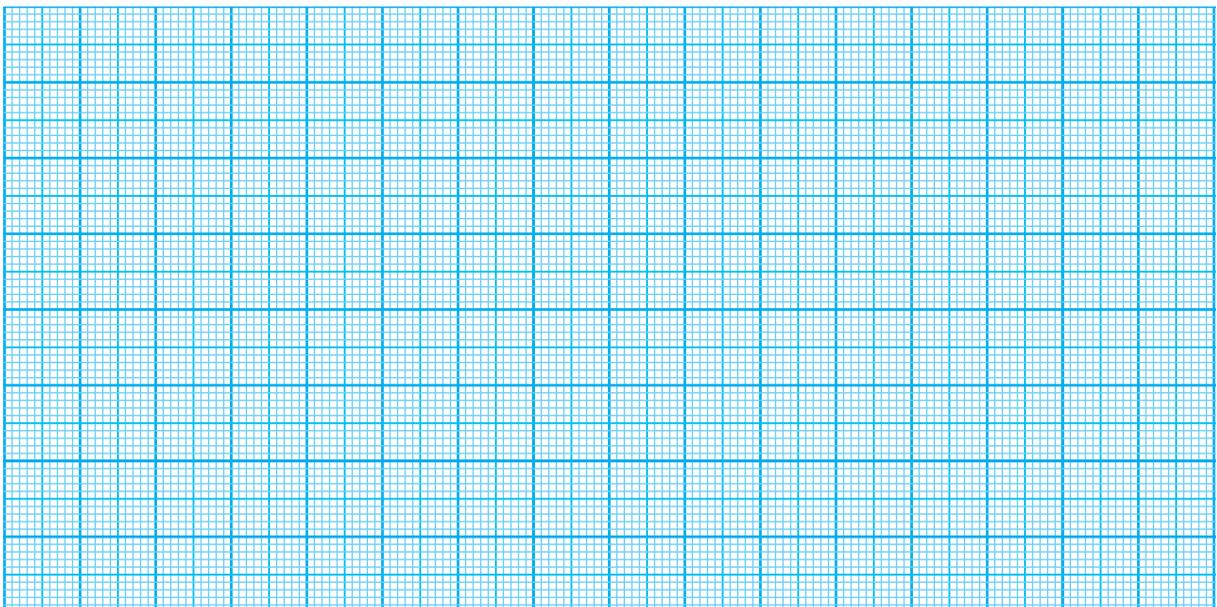
Remarque : Dans certains cas l'asservissement PID de position est inapplicable. En effet la masse doit pouvoir entraîner le moteur par inertie. Ce n'est pas le cas pour une transmission démultipliée par vis sans fin si cette dernière possède une inclinaison de denture inférieure à



- 1- Mettre chacun des boutons P, I et D sur la valeur zéro. Modifier la position désirée de la masse et observer le comportement du système. Tracer l'allure de la courbe qui représente la position de la masse dans le temps.



- 2- Choisir des valeurs arbitraires pour P, I et D. Introduire une perturbation sur le système. Tracer de nouveau l'allure de la courbe qui représente la position de la masse dans le temps. Comparer avec l'allure de la courbe précédente.





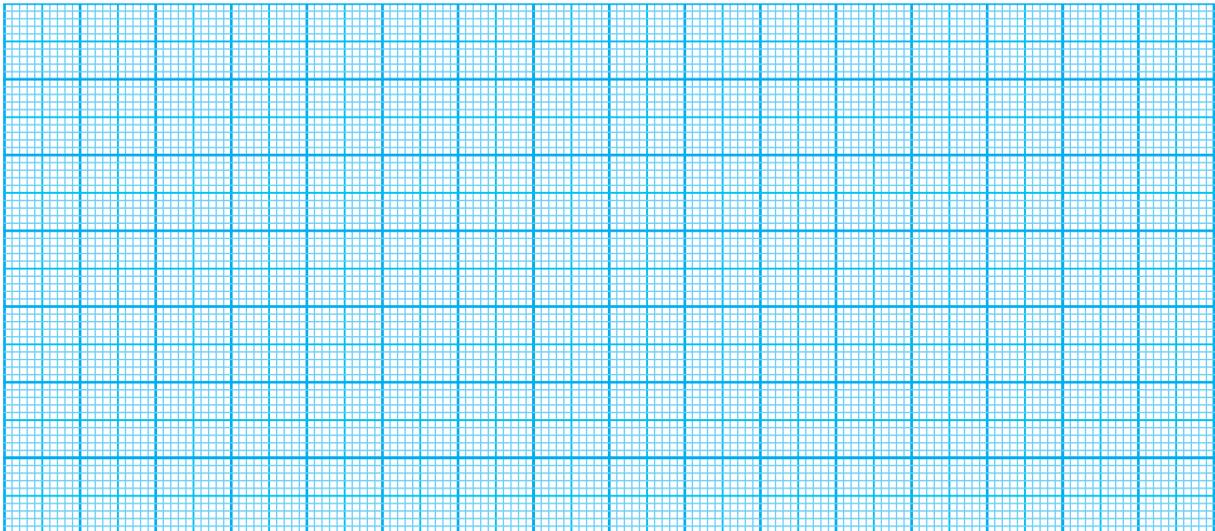
Activité 3

Il s'agit d'utiliser le logiciel " Correcteur PID " pour :

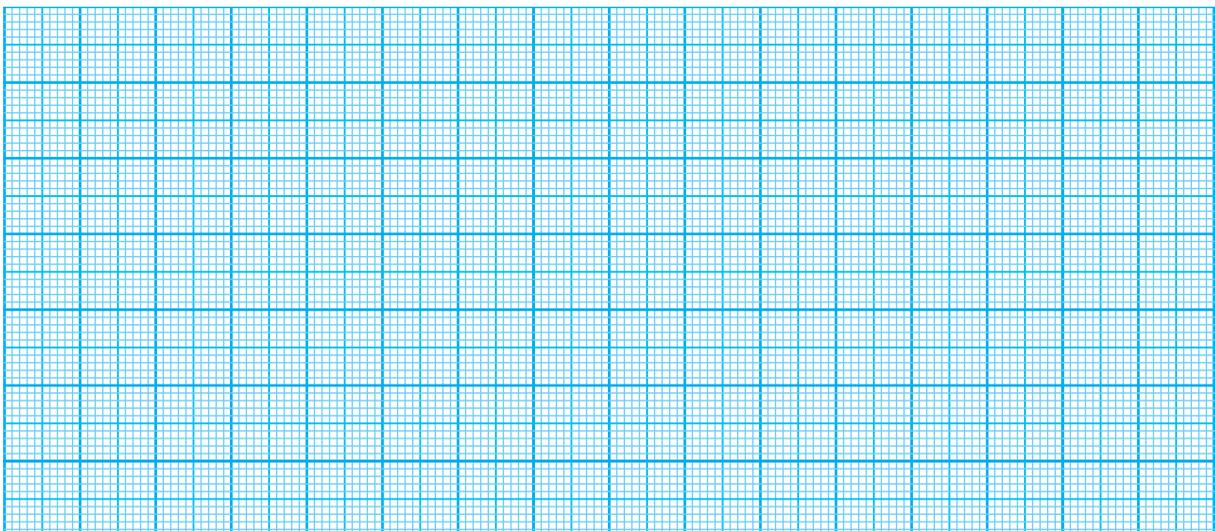
- ✓ mettre en évidence le comportement d'une boucle d'asservissement (brancher et débrancher la boucle)
- ✓ modifier les actions du correcteur et analyser les performances vis à vis d'une perturbation.

Utiliser le **logiciel** de simulation "**WPID**" pour simuler le comportement d'un système asservi en position en activant les différents correcteurs disponibles à savoir **P**, **I** et **D**.

- 1- Mettre les boutons **I** et **D** sur la valeur zéro. Choisir quelques valeurs différentes pour **P**, modifier la consigne de position et observer le comportement du système. Tracer pour deux valeurs de **P** l'allure de la courbe qui représente la position de la masse dans le temps.

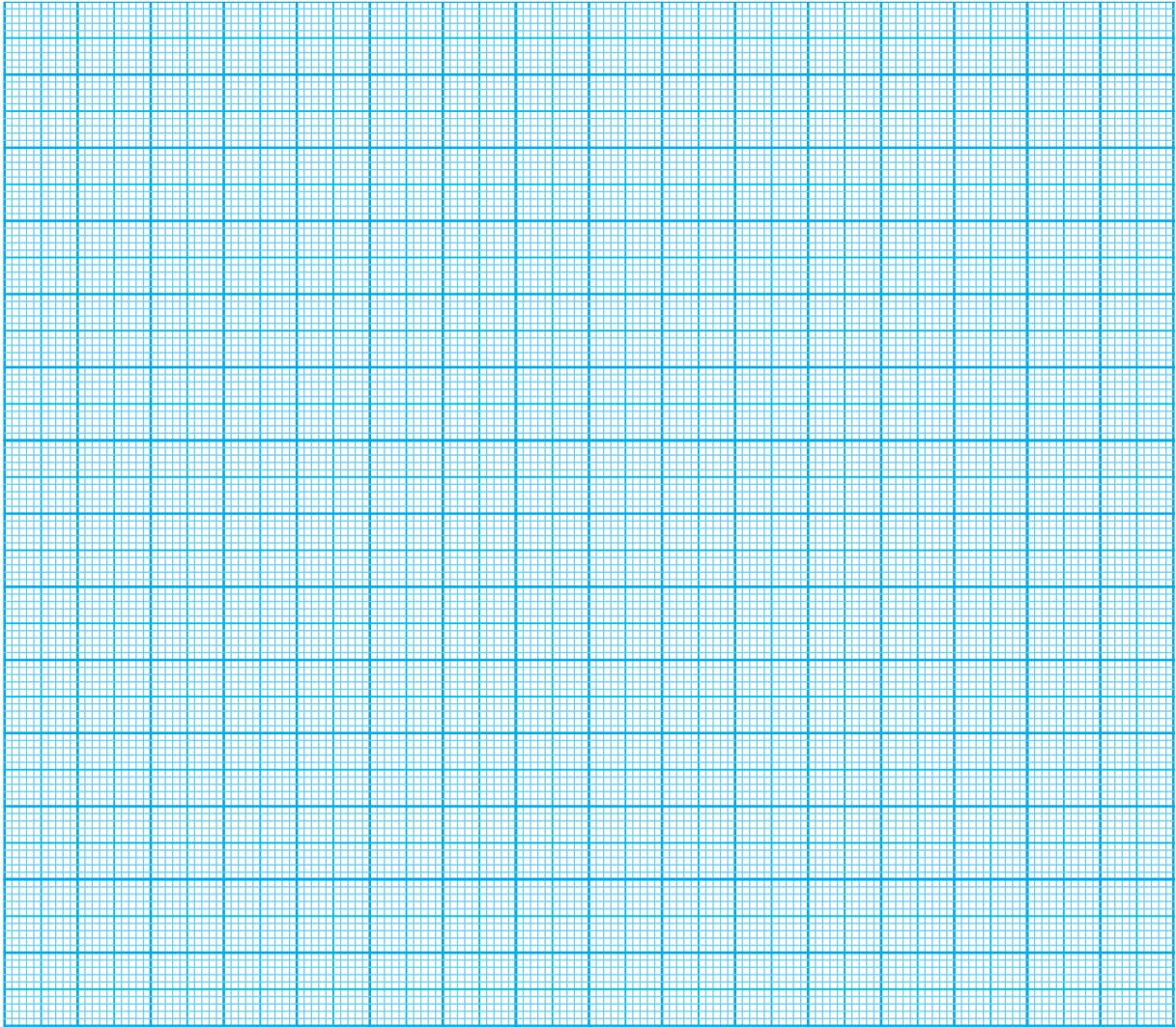


- 2- Mettre les boutons **P** et **D** sur la valeur zéro. Choisir quelques valeurs différentes pour **I**, modifier la consigne de position et observer le comportement du système. Tracer pour deux valeurs de **I** l'allure de la courbe qui représente la position de la masse dans le temps.





3- Mettre les boutons **P** et **I** sur la valeur zéro. Choisir quelques valeurs différentes pour **D**, modifier la consigne de position et observer le comportement du système. Tracer pour deux valeurs de **D** l'allure de la courbe qui représente la position de la masse dans le temps.



4- A partir des résultats obtenus précédemment, remplir le tableau suivant en mettant un signe + pour amélioration et un signe - pour la dégradation.

	Précision	Stabilité	Rapidité
P			
I			
D			

TP B1

Les Systèmes Triphasés

- Objectifs Spécifiques :

OS B1-2 - Déterminer les grandeurs électriques d'un réseau triphasé.

OS B1-3 - Déterminer les grandeurs électriques d'un réseau triphasé équilibré.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Moteur asynchrone monophasé
- ✓ Moteur asynchrone triphasé
- ✓ Appareils de mesure
- ✓ Rhéostat de charge variable.

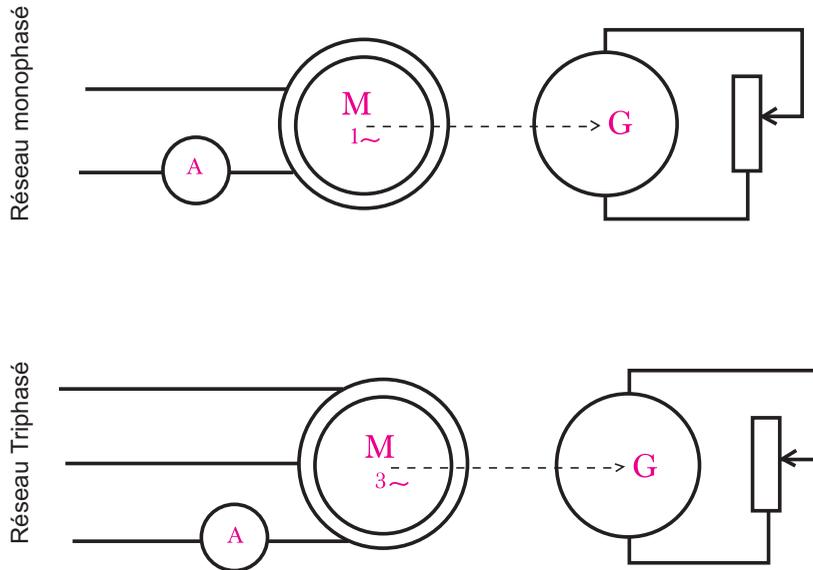


Activité 1

Il s'agit d'accoupler une même charge :

- ✓ à un moteur asynchrone triphasé
- ✓ à un moteur asynchrone monophasé

de mesurer dans les deux cas le courant appelé par ligne et de comparer les résultats.



$I_{mono} = \dots\dots\dots$

$I_{tri} = \dots\dots\dots$

TP B2

Les Moteurs Asynchrones Triphasés

- Objectifs Spécifiques :

OS B3-1 - Identifier un moteur asynchrone triphasé à cage.

OS B3-2 - Mettre en oeuvre un moteur asynchrone à cage.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Machine didactique de démonstration (ou matériel équivalent)
- ✓ Lot d'appareils de mesure
- ✓ Variateur de vitesse
- ✓ Groupe de machine "Moteur asynchrone + charge variable".



Activité 1

Il s'agit de comparer le fonctionnement :

- d'une aiguille aimantée
- d'un disque en cuivre ou en aluminium placé au milieu de trois bobines alimentées en courant alternatif.

La première manipulation consiste à placer une aiguille aimantée au centre de gravité d'un système de 3 bobines disposées à 120° géométrique sur un plan horizontal.

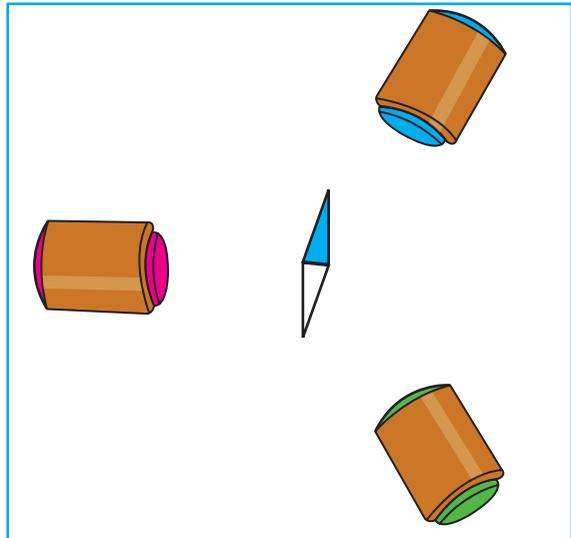
On alimente les 3 bobines par un réseau triphasé (de préférence à faibles tensions).

1- Que peut-on constater ?

.....

2- Mesurer la vitesse qu'on appellera Ω_s par la méthode stroboscopique

$$\Omega_s = \dots\dots\dots \text{tr/s} = \dots\dots\dots \text{tr/mn}$$



La manipulation consiste à placer un disque métallique (aluminium, cuivre ou alliage léger) au centre de gravité d'un système de 3 bobines disposées à 120° géométriques sur un plan horizontal.

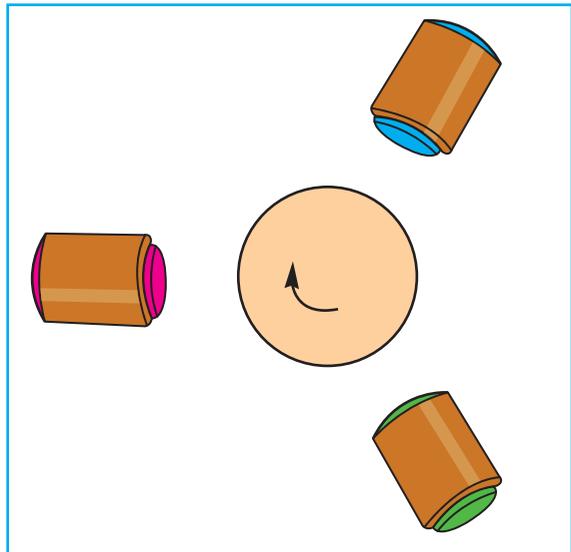
On alimente les 3 bobines par un réseau triphasé (de préférence à faibles tensions).

1- Que peut-on constater ?

.....

2- Mesurer la vitesse qu'on appellera Ω par la méthode stroboscopique

$$\Omega = \dots\dots\dots \text{tr/s} = \dots\dots\dots \text{tr/mn}$$



3- Conclure :

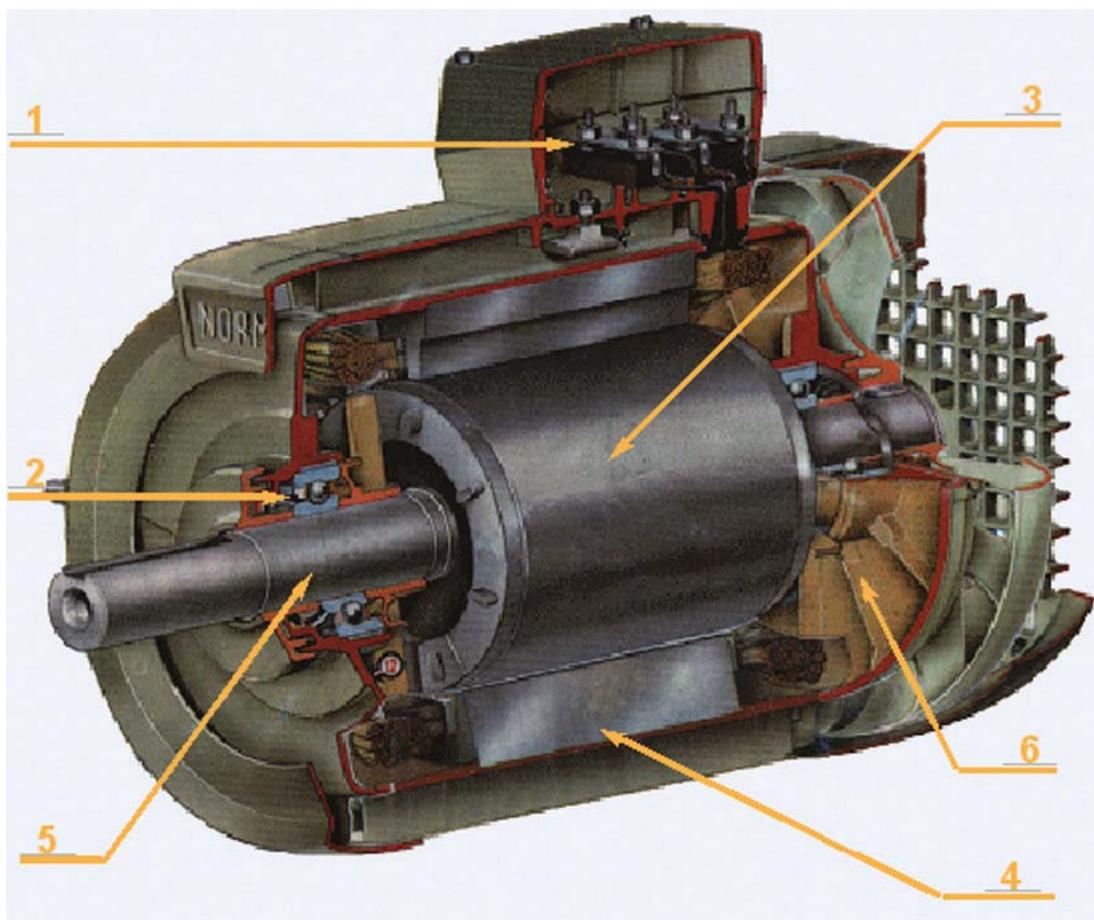
.....



Activité 2

Il s'agit d'identifier sur une photo en coupe partielle les différents éléments qui constituent un moteur asynchrone à rotor en court-circuit :

Compléter le tableau suivant par le repère qui correspond à chaque désignation.



Désignation	Repère	Désignation	Repère
Le rotor		Le stator	
L'arbre		La plaque à bornes	
Le roulement à billes		Le ventilateur	



Activité 3

Il s'agit de déterminer pour un moteur asynchrone tournant à vide :

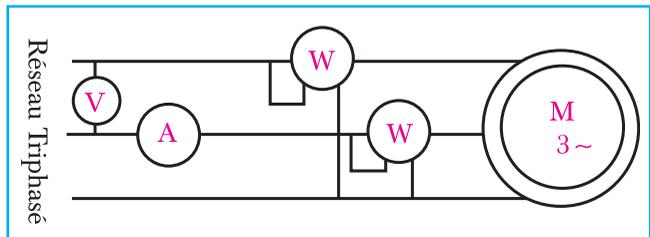
- le courant à vide
- les pertes constantes
- la vitesse de rotation et le glissement.

Réaliser le montage de la figure suivante et faire tourner le moteur sans aucune charge

Relever alors les valeurs :

- ✓ du courant à vide
- ✓ de la puissance mesurée par les deux wattmètres
- ✓ de la tension composée
- ✓ la vitesse de rotation.

I_0	U_0	P_{10}	P_{20}	n_0



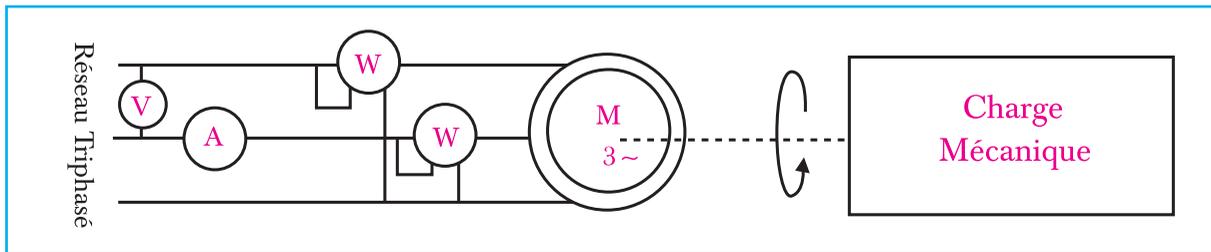
Activité 4

Il s'agit de déterminer pour un moteur asynchrone tournant en charge :

- la vitesse de rotation et le glissement
- la puissance active absorbée par le moteur
- les pertes joules statoriques
- la puissance transmise
- le couple électromagnétique
- la puissance utile.

Réaliser le montage de la figure suivante et faire tourner le moteur accouplé à une génératrice débitant sur un rhéostat. Faire varier la charge de la génératrice pour modifier le couple résistant et relever alors pour chaque cas :

- ✓ le courant de ligne
- ✓ la puissance mesurée par les deux wattmètres
- ✓ la tension composée
- ✓ la vitesse de rotation



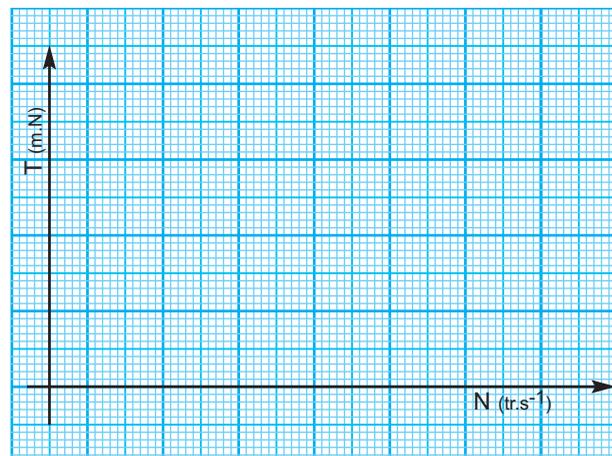
Calculer dans ce cas :

- ✓ le glissement
- ✓ la puissance active absorbée par le moteur
- ✓ les pertes joules statoriques
- ✓ la puissance transmise
- ✓ le couple électromagnétique
- ✓ la puissance utile
- ✓ le rendement par la méthode directe puis indirecte ; et compléter le tableau.

Remarque : La résistance d'un enroulement du stator est mesurée à chaud juste après l'arrêt du moteur. Cette mesure est effectuée sous tension continue réduite.

I	U	P ₁	P ₂	n	Pa	g	P _{js}	P _{tr}	Te	Pu	η

Tracer sur le papier millimétré ci-contre la courbe qui représente la variation du couple en fonction de la vitesse de rotation (se limiter à la partie linéaire).



TP B3

Les Moteurs à Courant Continu

- Objectifs Spécifiques :

OS B4-1 - Tracer les caractéristiques d'un moteur à courant continu.

OS B4-2 - Déterminer le point de fonctionnement pour une charge donnée.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Machine didactique de démonstration (ou matériel équivalent)
- ✓ Lot d'appareils de mesure
- ✓ Variateur de vitesse
- ✓ Groupe de machine "Moteur à courant continu + charge variable"



Activité 1

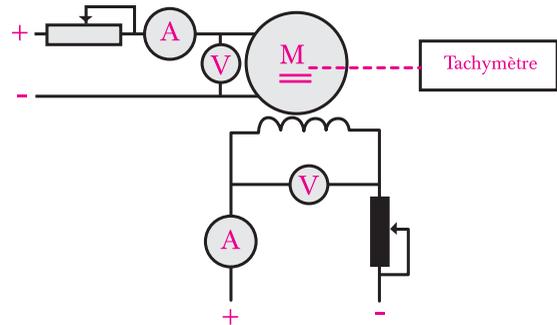
Il s'agit de faire tourner à vide un moteur à courant continu et de mesurer :

- ✓ la vitesse de rotation
- ✓ l'intensité du courant absorbé par l'induit
- ✓ l'intensité du courant absorbé par l'inducteur
- ✓ la tension appliquée aux bornes de l'induit
- ✓ la tension appliquée aux bornes de l'inducteur.

1- Réaliser le montage de la figure suivante et faire tourner le moteur sans aucune charge.

2- Relever alors les valeurs :

- ✓ du courant à vide absorbé par l'induit
- ✓ de la tension appliquée à l'induit
- ✓ de la tension appliquée à l'inducteur
- ✓ du courant d'excitation
- ✓ la vitesse de rotation

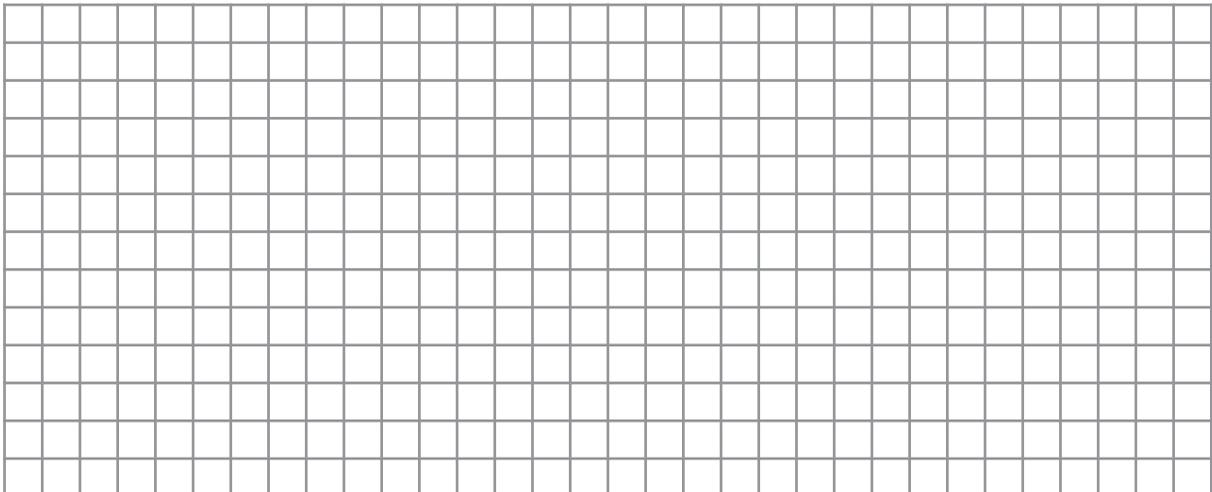


I_a	U_a	i_{ex}	u_{ex}	n_0

3- Mesurer la résistance de l'induit à chaud sous très faible tension continue

4- Calculer :

- ✓ les pertes joules inducteur
- ✓ les pertes joules induit
- ✓ la puissance totale absorbée à vide par le moteur
- ✓ en déduire les pertes constantes
- ✓ la force contre-électromotrice E' et en déduire le coefficient de proportionnalité entre E' et la vitesse de rotation.



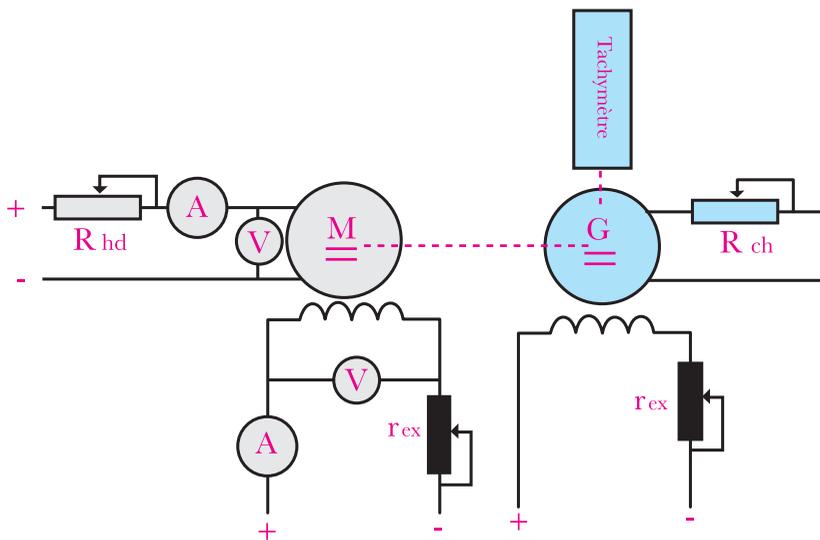


Activité 3

Il s'agit de faire tourner un moteur à courant continu, de faire varier sa charge et de mesurer pour différentes situations :

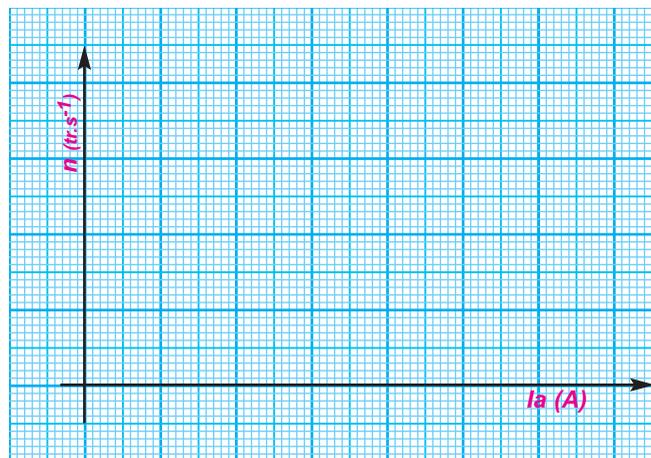
- ✓ la vitesse de rotation
- ✓ l'intensité du courant absorbé par l'induit
- ✓ le couple fourni par le moteur (par acquisition numérique ou par mesure avec dynamo-balance).

- 1- Réaliser le montage de la figure suivante et faire tourner le moteur accouplé à une charge variable.
- 2- Relever alors pour **différentes charges** et pour un **courant d'excitation constant** et pour une **tension** appliquée à l'induit **constante**, les valeurs :
 - ✓ du courant absorbé par l'induit
 - ✓ la vitesse de rotation



I_a	U_a	n

- 3- Tracer sur le repère donné ci-contre, la courbe de la vitesse de rotation n en fonction du courant absorbé par l'induit I_a : $n = f(I_a)$

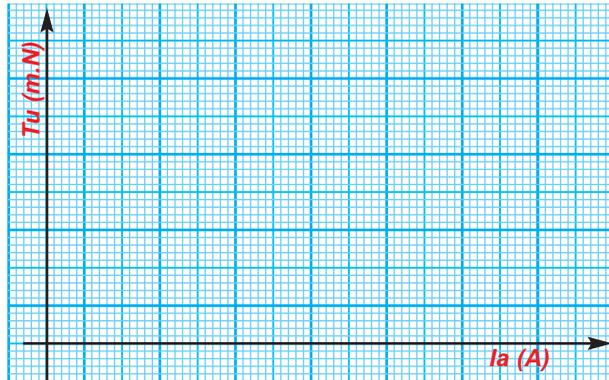




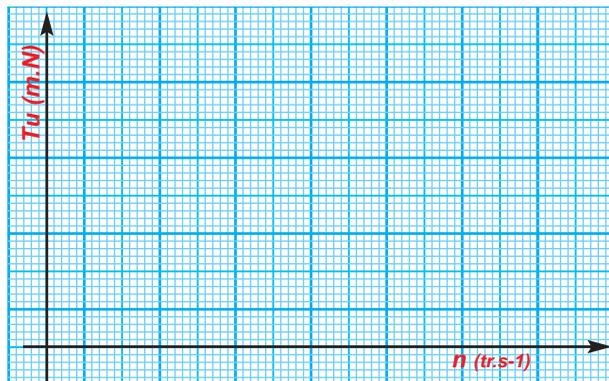
- 4- Relever ensuite dans un autre essai, pour **différentes charges**, pour une **vitesse constante** et pour un **courant d'excitation constant**, les valeurs :
- du courant absorbé par l'induit
 - du couple développé par l'arbre du moteur.

I_a	U_a	T_u

- 5- Tracer sur le repère donné ci-contre, la courbe du couple utile **Tu** en fonction du courant absorbé par l'induit **Ia** : **Tu = f(Ia)**



- 6- En déduire sur le repère donné ci-contre, la courbe du couple utile **Tu** en fonction de la vitesse de rotation **n** : **Tu = f(n)**





Activité 4

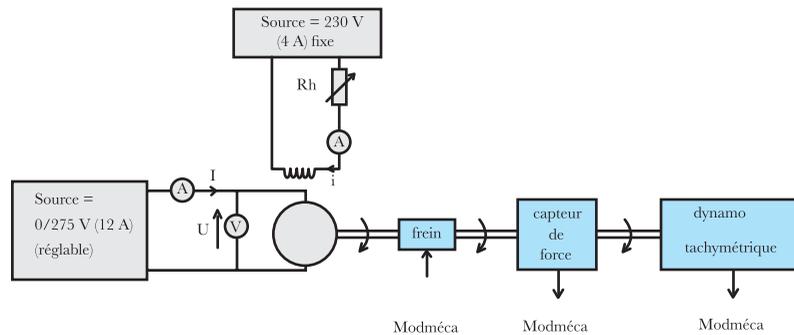
Il s'agit de faire tourner un moteur à courant continu, de faire varier sa vitesse et de mesurer pour différentes situations :

- ✓ la vitesse de rotation ;
- ✓ l'intensité du courant absorbé par l'induit
- ✓ le couple fourni par le moteur (par acquisition numérique ou par mesure avec dynamo-balance)

1- Réaliser le montage de la figure ci-contre :

2- Alimenter l'inducteur, diminuer R_h jusqu'à $i_{nominal}$.

3- Démarrer le moteur à courant continu sous tension variable.



NB : a) Pour arrêter le moteur :

- ✓ ramener T_u à 0 % .
- ✓ ramener U à 0 V.
- ✓ ramener R_h à sa valeur **max**.

b) Les grandeurs sur lesquelles on peut agir sont :

- ✓ i : par l'intermédiaire de R_h .
- ✓ U : par l'intermédiaire de la source = réglable.
- ✓ $Tr (=T_u)$ par la commande du frein.

4- Le moteur démarré, maintenir $i = cte$, $U = cte$, et agir sur la commande du frein (T_u manu) pour faire varier la charge, donc T_u , de 0 à 10Nm. Relever alors les valeurs de :

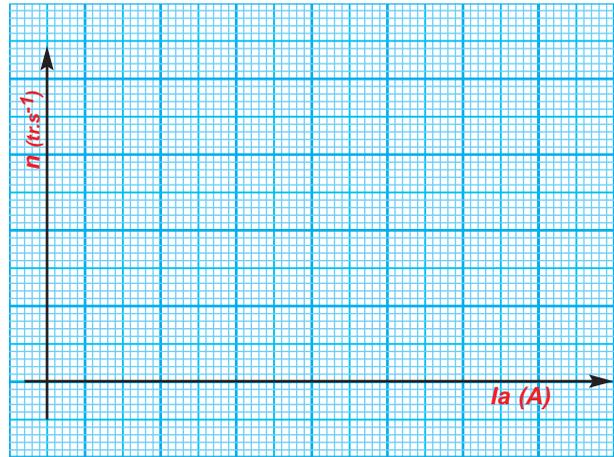
I_a : Modélec

Modméca

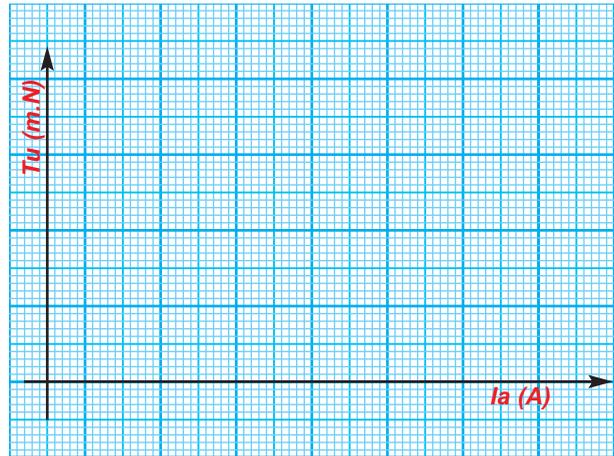
I_a	P_u	n	T_u



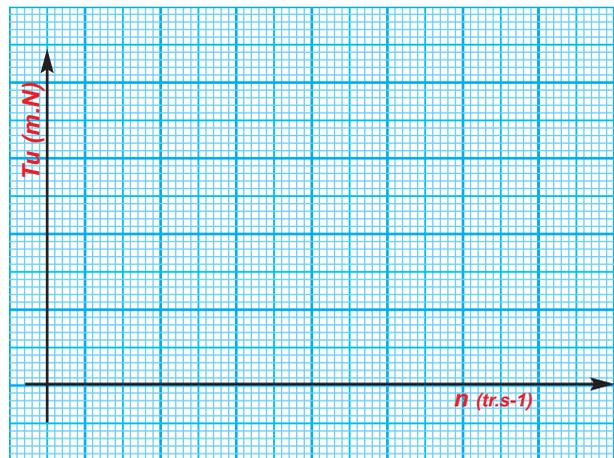
- 5- Tracer sur le repère donné ci-contre, la courbe de la vitesse de rotation n en fonction du courant absorbé par l'induit I_a : $n = f(I_a)$



- 6- Tracer sur le repère donné ci-contre, la courbe du couple utile T_u en fonction du courant absorbé par l'induit I_a : $T_u = f(I_a)$



- 7- En déduire sur le repère donné ci-contre, la courbe du couple utile T_u en fonction de la vitesse de rotation n : $T_u = f(n)$



TP C1

Les Amplificateurs Linéaires Intégrés

- Objectifs Spécifiques :

OS C1-1 - Identifier les différents types des montages mettant en oeuvre des A.L.I sur schéma structurel.

OS C1-2 - Réaliser et mettre en oeuvre des montages à base d'A.L.I.

- Conditions de réalisation :

Les activités sont organisées par groupe de 4 ou 5 élèves

Exemple d'équipements utilisés :

- ✓ Poste PC
- ✓ Logiciel de simulation analogique installé (ORCAD)
- ✓ Matériel d'essai en électronique
- ✓ Simulateur analogique.



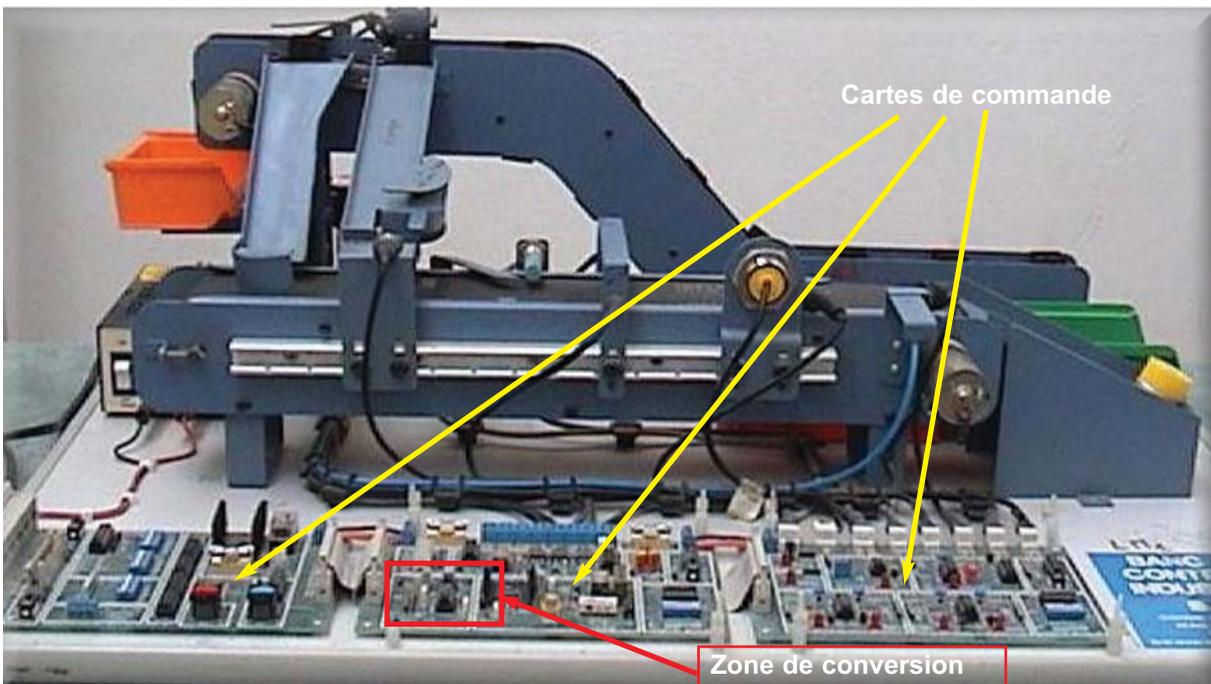
Activité 1

Il s'agit, pour un amplificateur linéaire, d'identifier sur le schéma structurel de la carte de commande du système de tri S.E.T :

- ✓ le régime de fonctionnement
- ✓ la fonction qu'il assure.

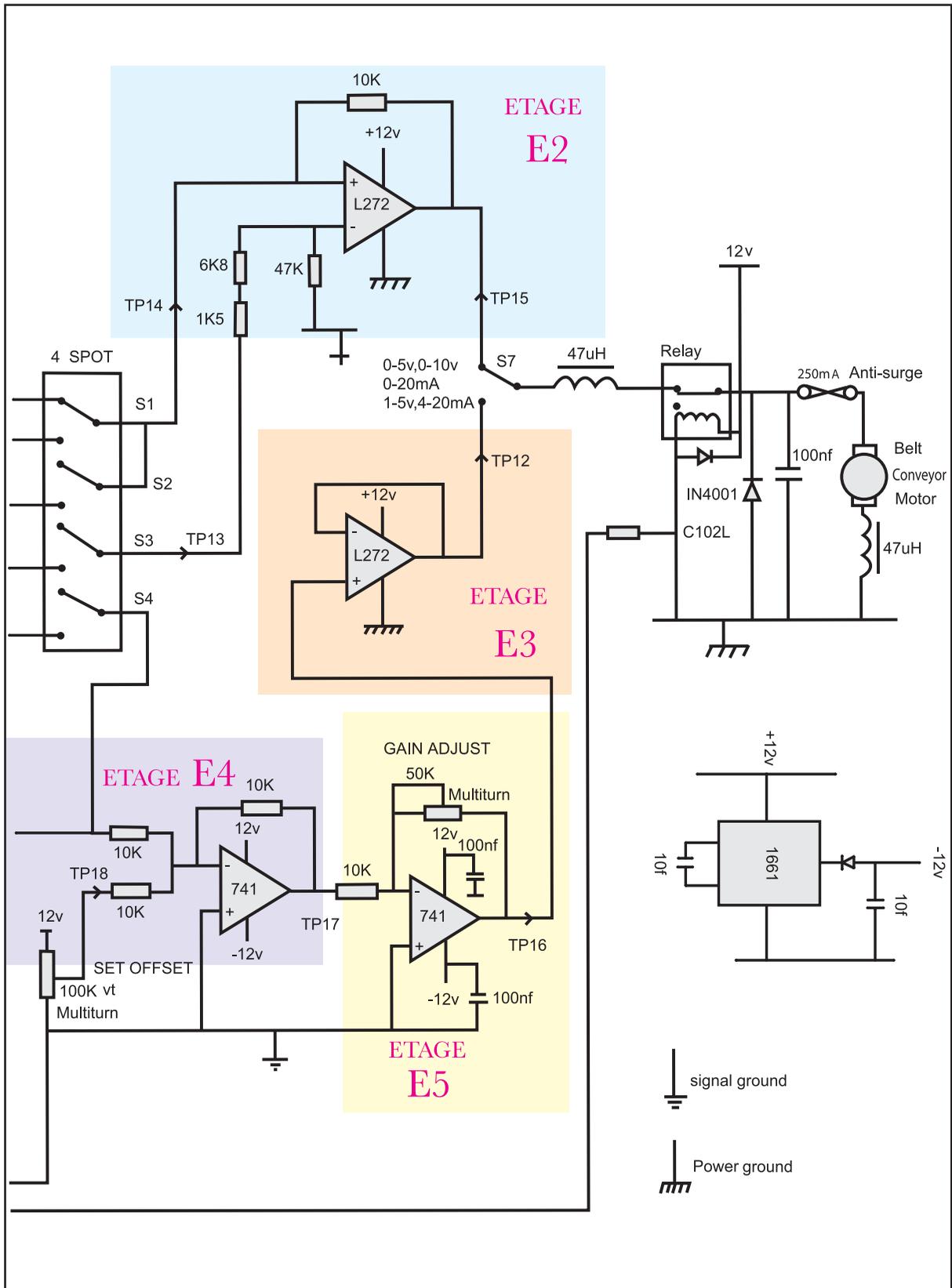
La maquette du système de tri S.E.T exploite plusieurs types de cartes de commande réelles. En effet la communication entre l'ordinateur et le système nécessite des circuits pour l'adaptation des signaux, la mise en forme et la conversion.

La figure suivante illustre l'emplacement des trois cartes de commande utilisées pour piloter le système S.E.T par automate programmable ou par ordinateur équipé d'une carte d'interface et du logiciel approprié.



- 1- En se référant au schéma structurel de la carte de commande du système de tri S.E.T. Compléter le tableau suivant :

Bloc	Boucle (ouverte, fermée)	Régime de fonctionnement	Equation de la sortie	Fonction réalisée par le bloc
E3				
E4				
E5				





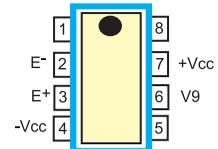
Activité 2

Il s'agit pour un amplificateur linéaire fonctionnant en boucle ouverte, de tracer l'allure de la tension de sortie en fonction des entrées :

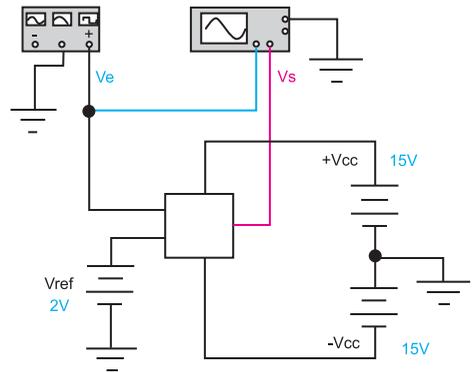
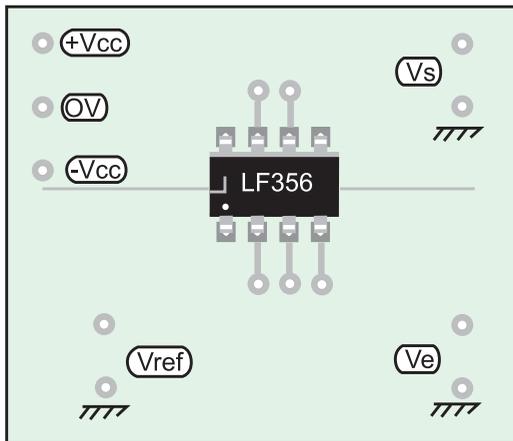
- ✓ pour une polarisation symétrique
- ✓ pour une polarisation asymétrique.

Avec :

- ✓ $V_{ref} = 2V$ tension de référence.
- ✓ $V_e(t)$ tension d'entrée triangulaire alternative d'amplitude 3V et de fréquence $f=100$ Hz ;
- ✓ $|+V_{cc}| = |-V_{cc}| = 12V$ tensions de polarisation.

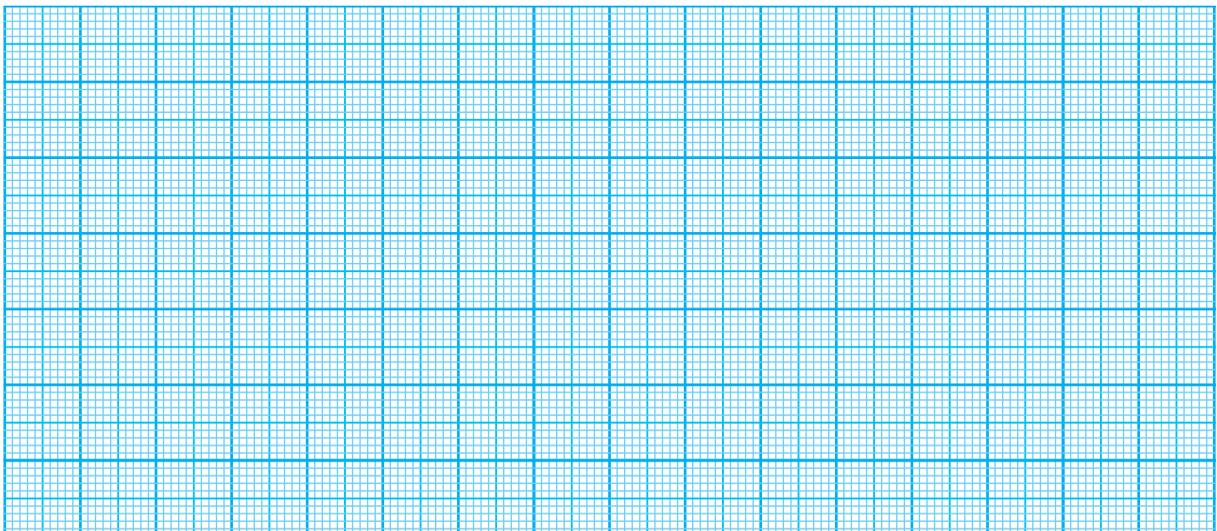


Brochage de l'A.L.I



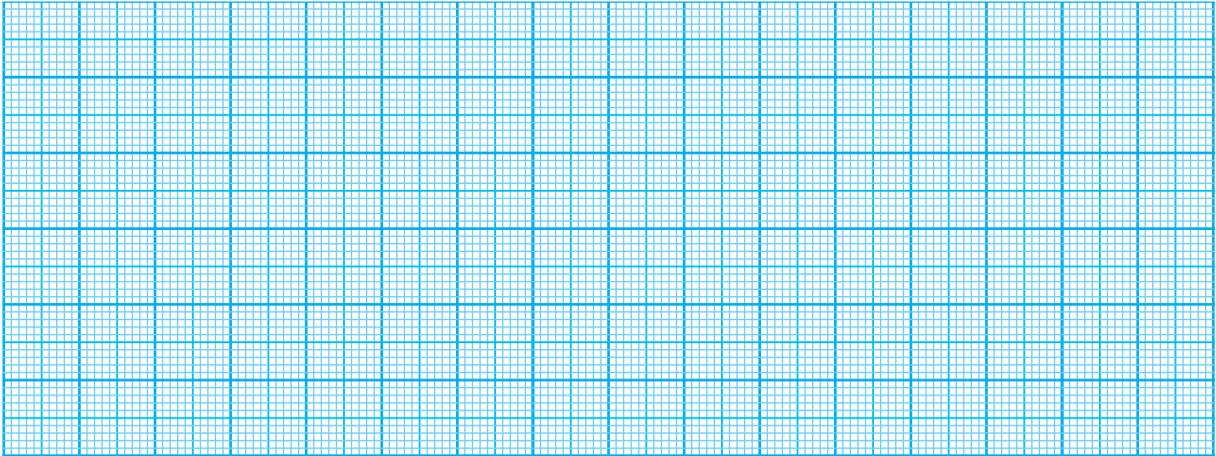
1- Polariser le circuit amplificateur entre +12V et la masse puis :

- ✓ Visualiser et tracer $V_e(t)$ et $V_s(t)$ sur le même graphe.



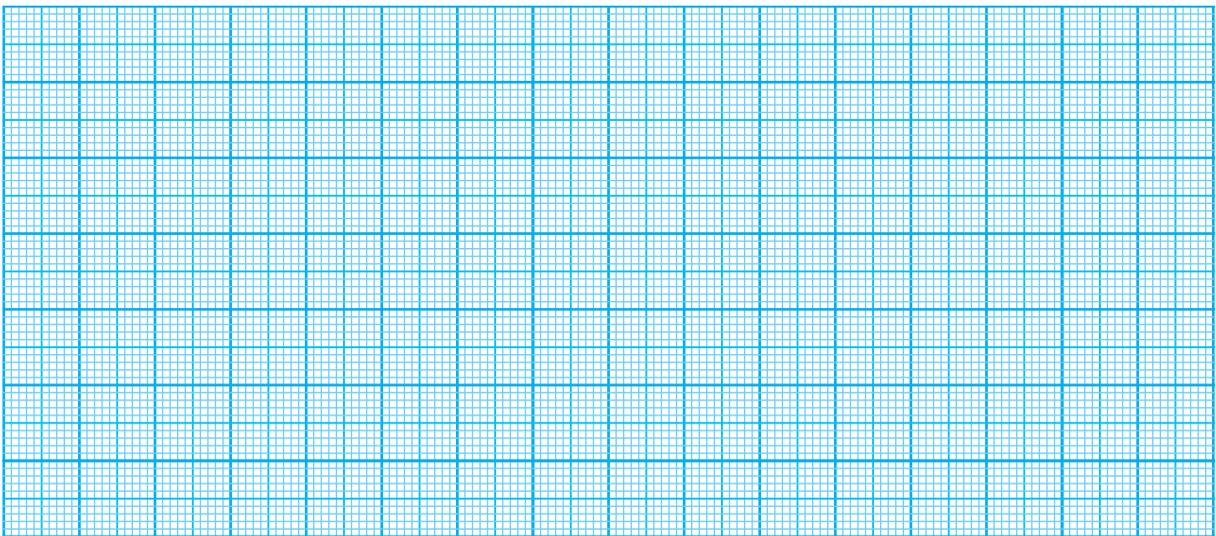


✓ Déduire de ces tensions, la caractéristique $V_s = f(V_e)$ avec $f = 100$ Hz.

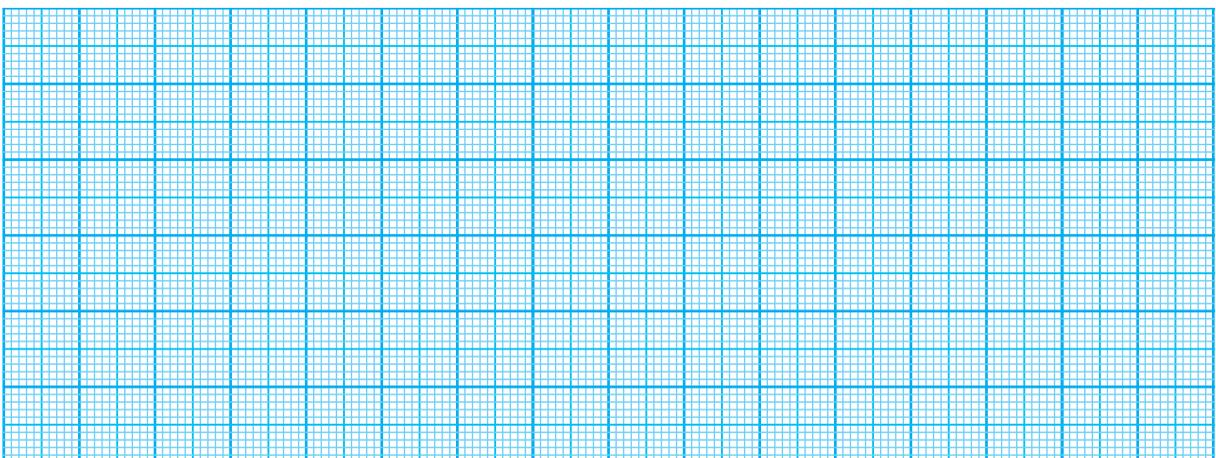


2- Polariser le circuit amplificateur entre +12V et -12V puis :

✓ visualiser et tracer $V_e(t)$ et $V_s(t)$ sur le même graphe.



✓ Déduire de ces tensions, la caractéristique $V_s = f(V_e)$ avec $f = 100$ Hz.





Activité 3

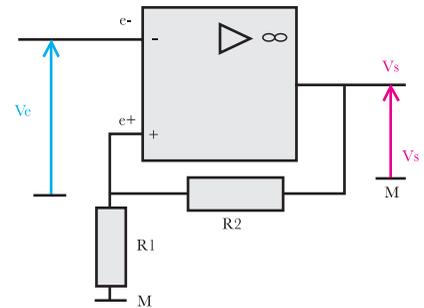
Il s'agit de tracer la caractéristique d'un amplificateur linéaire fonctionnant en :

- ✓ Comparateur inverseur à double seuils
- ✓ Comparateur non inverseur à double seuils

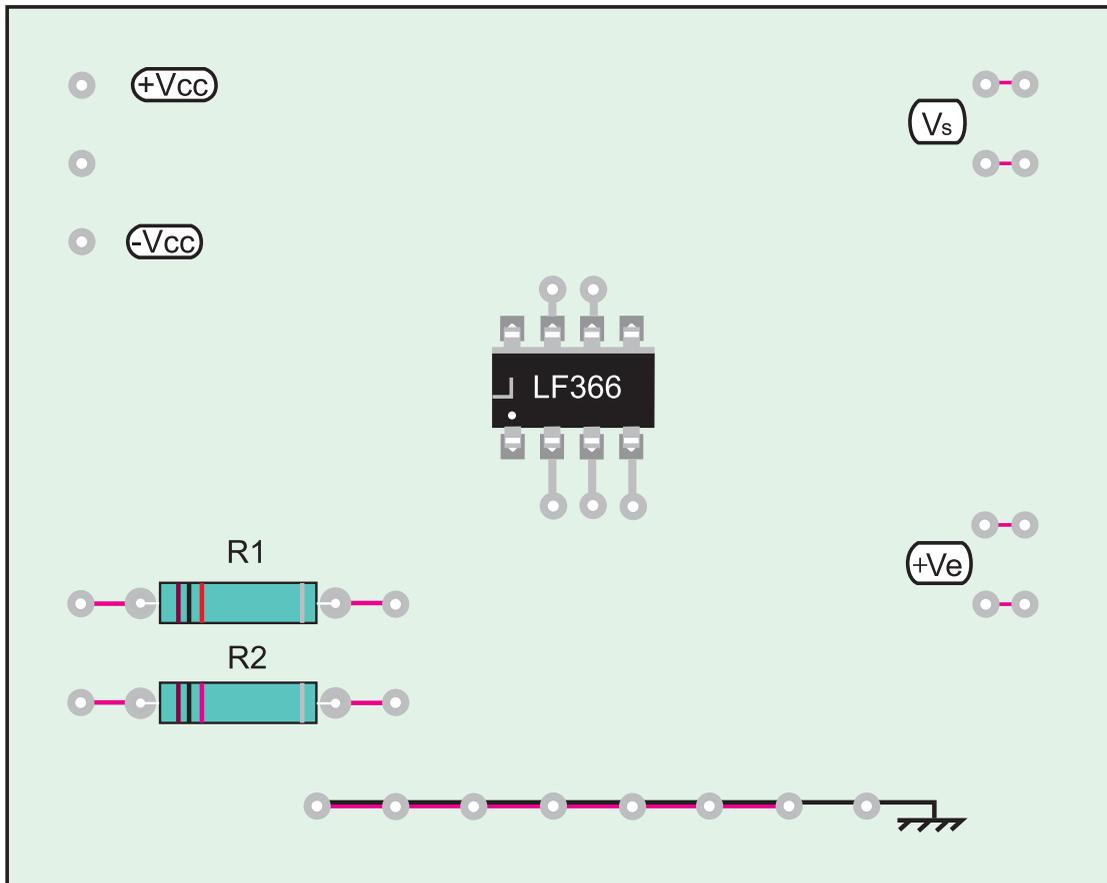
1- Fonctionnement d'un amplificateur linéaire en comparateur inverseur à double seuils

Avec :

- ✓ $V_e(t)$ tension d'entrée triangulaire alternative d'amplitude 3V et de fréquence $f=100$ Hz;
- ✓ $|+V_{cc}| = |-V_{cc}| = 12V$ tensions de polarisation.
- ✓ $R_1=1k\ \Omega$, $R_2=10k\Omega$.

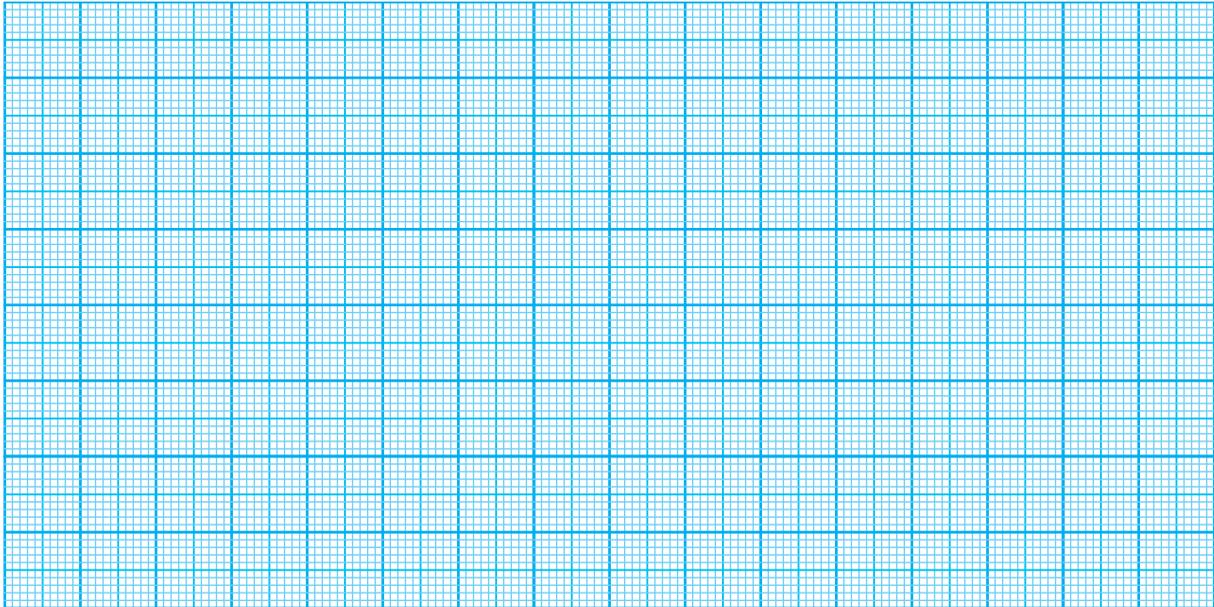


a. Réaliser sur la platine le schéma de câblage.

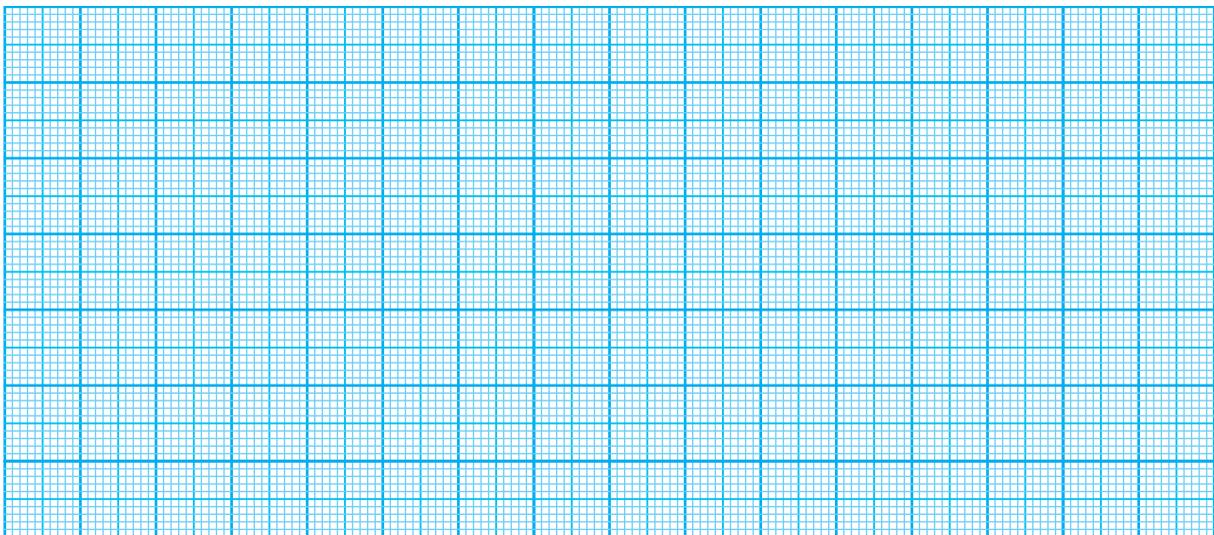




b. Visualiser et tracer $V_e(t)$ et $V_S(t)$ sur le même graphe .



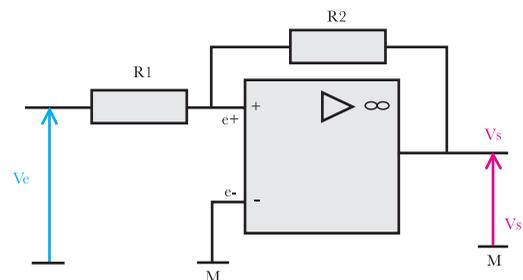
c. Déduire de $V_e(t)$ et $V_S(t)$ la caractéristique V_S en fonction de V_e : $V_S=f(V_e)$ avec $f = 1\text{Hz}$.



2- Fonctionnement d'un amplificateur linéaire en comparateur non inverseur à double seuils

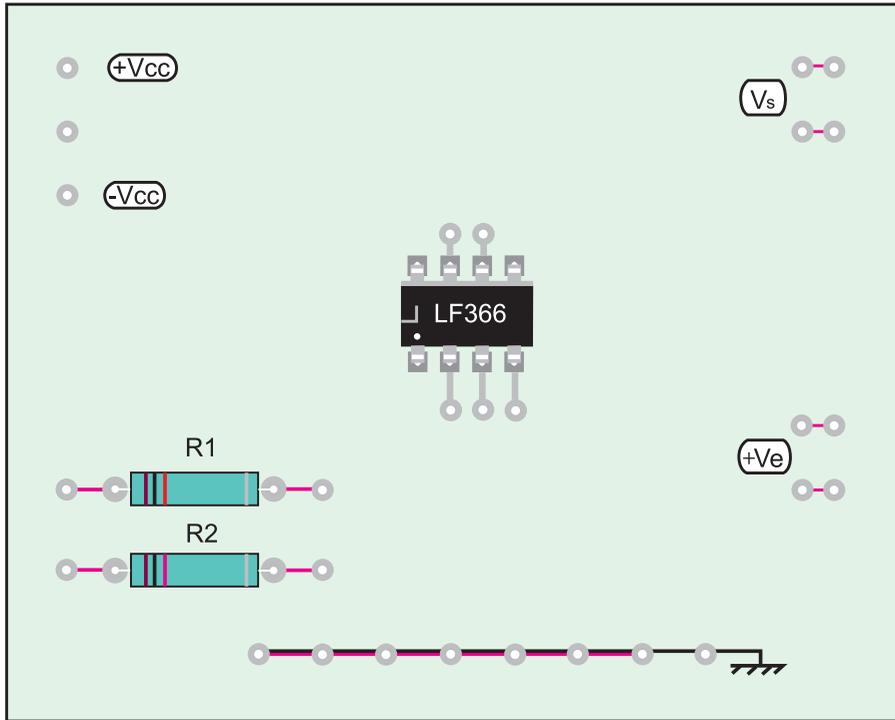
Avec :

- ✓ $V_e(t)$ tension d'entrée triangulaire alternative d'amplitude 3V et de fréquence $f=100\text{ Hz}$;
- ✓ $|+V_{cc}| = |-V_{cc}| = 12\text{V}$ tensions de polarisation.
- ✓ $R_1=1\text{k}\Omega$, $R_2=10\text{k}\Omega$

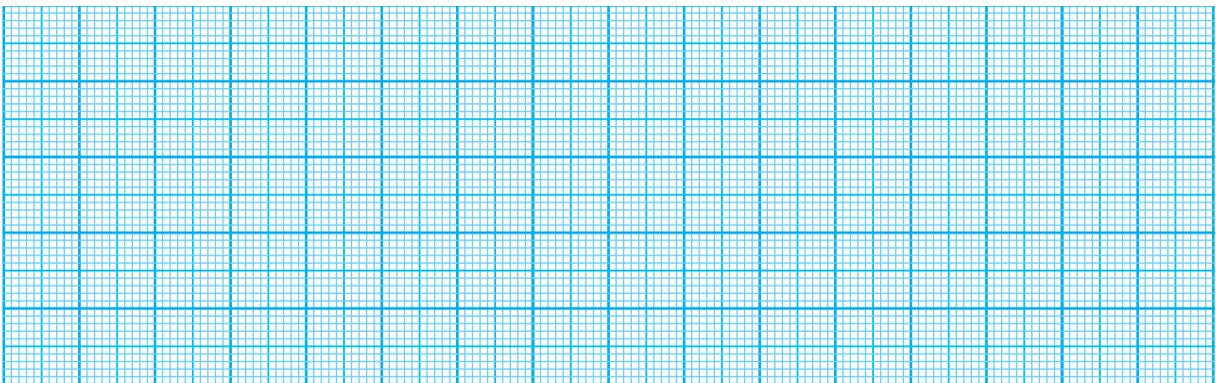




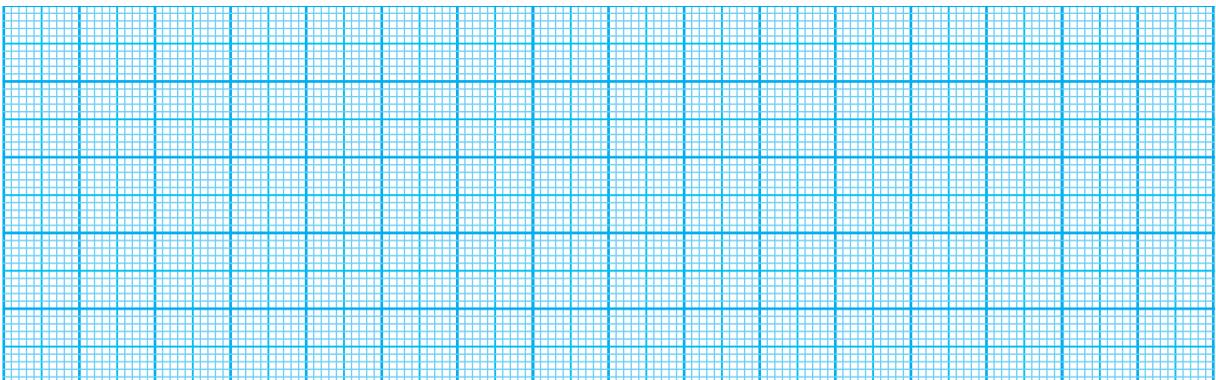
a. Réaliser sur la platine le schéma de câblage



b. Visualiser et tracer $V_e(t)$ et $V_s(t)$ sur le même graphe.



c. Déduire de $V_e(t)$ et $V_s(t)$ la caractéristique V_s en fonction de V_e : $V_s=f(V_e)$ avec $f = 1\text{Hz}$.





Activité 4

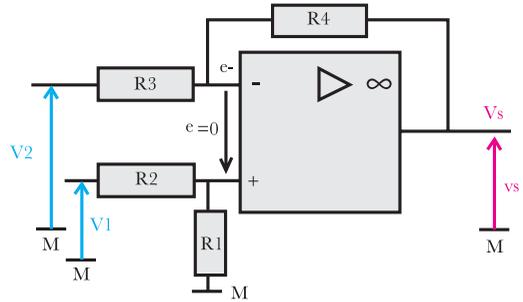
Il s'agit de tracer la courbe de sortie pour un amplificateur linéaire bouclé :

- ✓ sur un circuit résistif avec un double circuit d'attaque résistif
- ✓ sur un circuit capacitif avec un circuit d'attaque résistif
- ✓ sur un circuit résistif avec un circuit d'attaque capacitif.

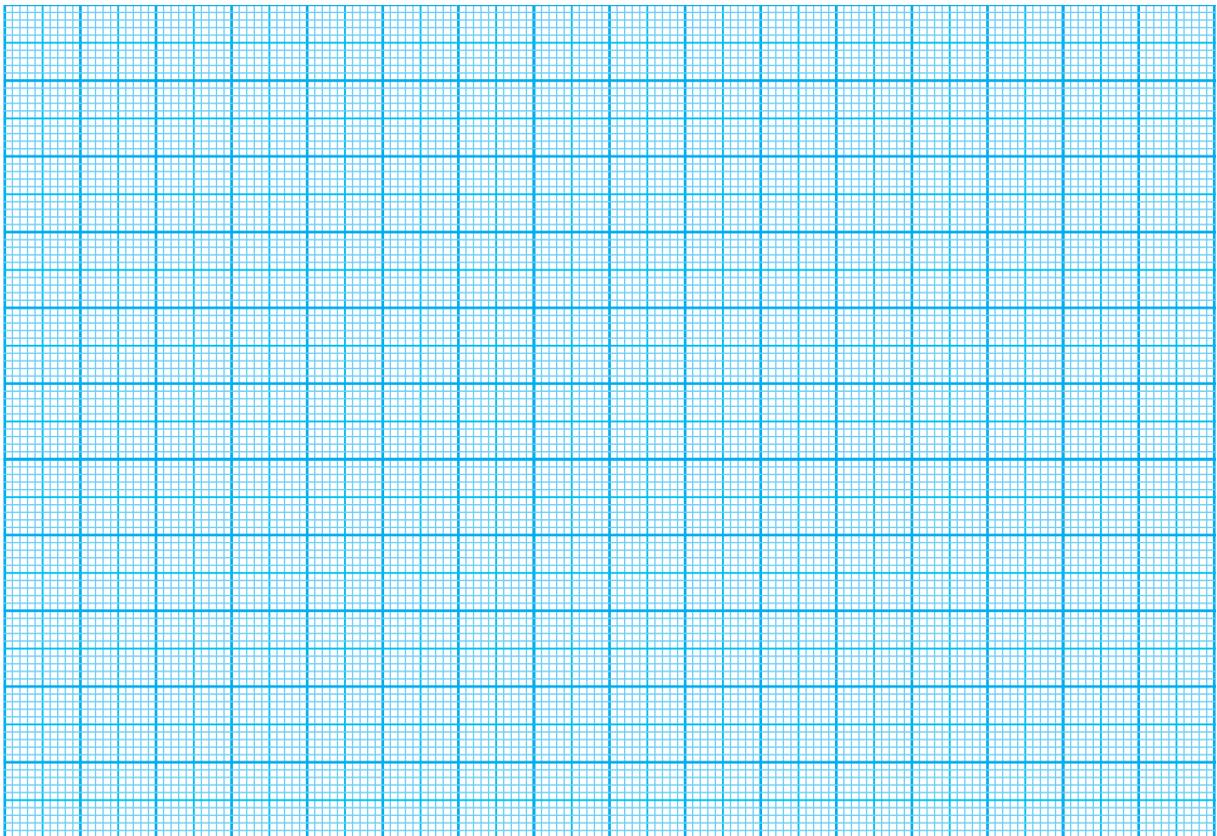
1- Fonctionnement en amplificateur de différence

Avec :

- ✓ $V_1(t)$ tension d'entrée triangulaire alternative d'amplitude 5V et de fréquence $f=100$ Hz ;
- ✓ $V_2(t)$ tension d'entrée triangulaire alternative d'amplitude 3V et de fréquence $f=100$ Hz ;
- ✓ $|+V_{cc}| = |-V_{cc}| = 12V$ tensions de polarisation.
- ✓ $R_1=R_2=R_3=R_4=1k\Omega$.



- a. Réaliser le schéma de câblage sur maquette, sur simulateur, ou avec un logiciel de simulation.
- b. Visualiser et tracer $V_1(t)$; $V_2(t)$ et $V_S(t)$ sur le même graphe.

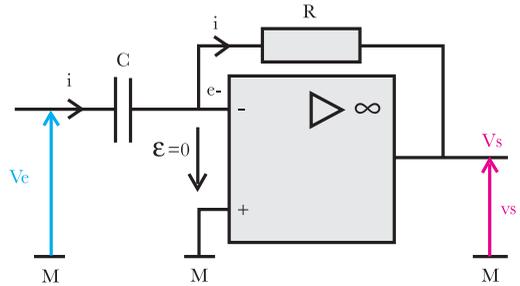




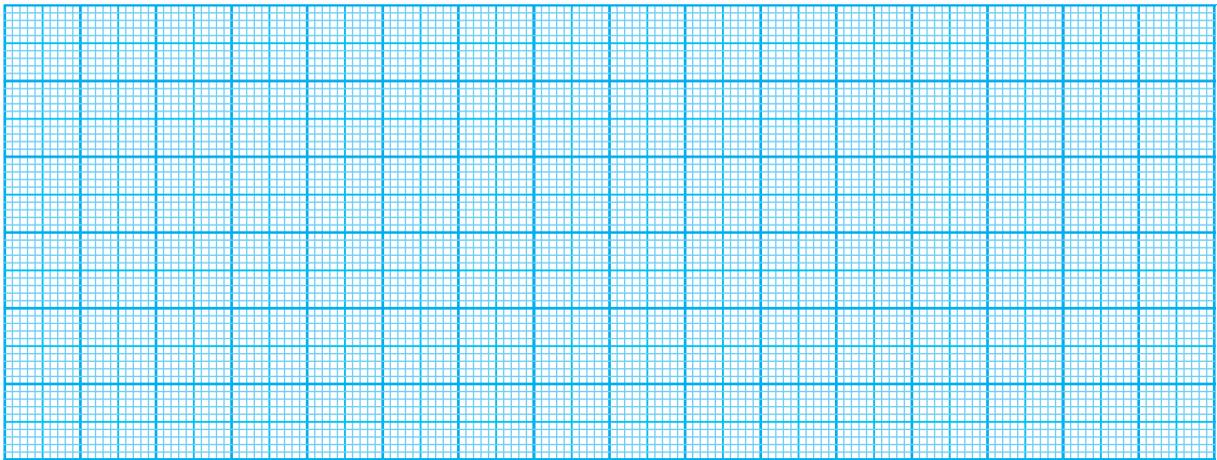
2- Fonctionnement en dérivateur :

Avec :

- ✓ $V_1(t)$ tension d'entrée triangulaire alternative d'amplitude 5V et de fréquence $f=100$ Hz;
- ✓ $|+V_{cc}| = |-V_{cc}| = 12V$ tensions de polarisation.
- ✓ $C = 1nF$; $R=1M\Omega$



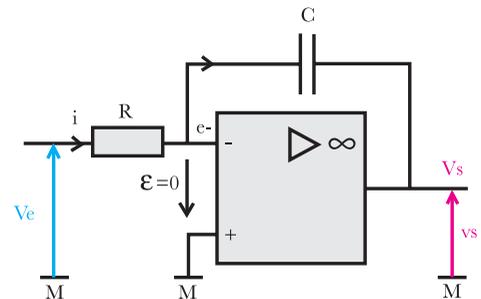
- a. Réaliser le schéma de câblage sur maquette, sur simulateur, ou avec un logiciel de simulation.
- b. Visualiser et tracer $V_e(t)$ et $V_s(t)$ sur le même graphe.



3- Fonctionnement en intégrateur

Avec :

- ✓ $V_1(t)$ tension d'entrée carrée alternative d'amplitude 5V et de fréquence $f=100$ Hz;
- ✓ $|+V_{cc}| = |-V_{cc}| = 12V$ tensions de polarisation.
- ✓ $C = 1nF$; $R=1M\Omega$



- a. Réaliser le schéma de câblage sur maquette, sur simulateur, ou avec un logiciel de simulation
- b. Visualiser et tracer $V_e(t)$ et $V_s(t)$ sur le même graphe.

