

# Le distributeur automatique de boissons

Site Internet :  
[www.gecif.net](http://www.gecif.net)

Type de document :  
**TP**

Intercalaire :

Date :

## I - Présentation du système

Le système que nous allons étudier ici est un distributeur automatique de boissons, qui rend la monnaie. Dans ce TP, nous nous intéressons essentiellement à la fonction « rendre la monnaie » réalisé par **le monnayeur**.

### I - 1 - Principe de fonctionnement

Ce distributeur automatique de boissons ne sait distribuer que 2 boissons différentes :

- \* de la menthe à l'eau
- \* du cassis à l'eau

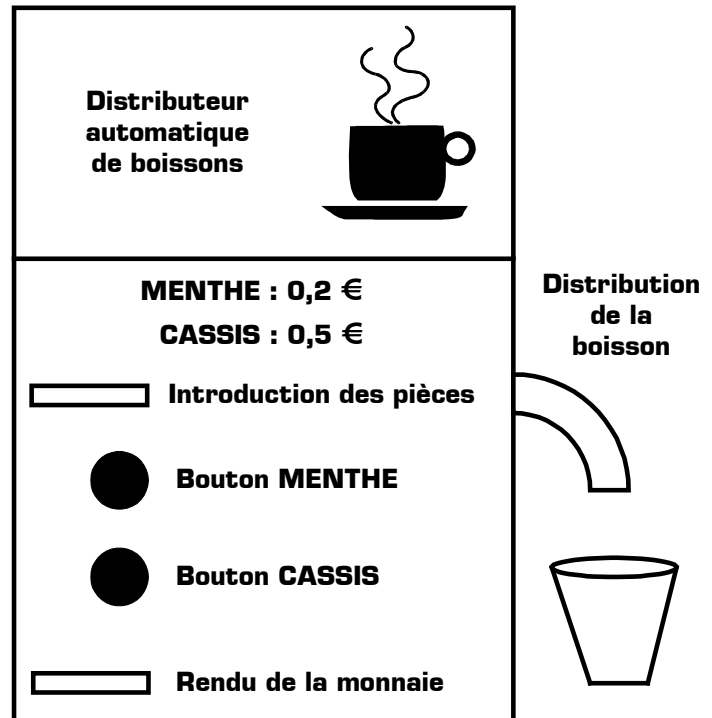
L'appareil dispose pour cela de 2 boutons sur sa face avant :

- \* un bouton **MENTHE** pour demander une menthe à l'eau
- \* un bouton **CASSIS** pour demander un cassis à l'eau

La menthe coûte 0,2 € et le cassis coûte 0,5 €

L'appareil n'accepte que 4 types de pièces de monnaie différentes :

- \* 0,10 €
- \* 0,20 €
- \* 0,50 €
- \* 1 €



Le monnayeur rend la monnaie si la somme introduite est supérieure au prix de la boisson demandée. Ce remboursement se fait de telle manière **qu'une seule pièce de chaque catégorie au maximum est rendue**. Par exemple, si l'utilisateur introduit une pièce de 1 € et qu'il demande une menthe à l'eau (prix 0,20 €), l'appareil doit rendre 0,80 €. Dans ce cas le distributeur de boissons ne va pas rendre 4 pièces de 0,20 € ni 8 pièces de 0,10 €, mais il rendra 1 pièce de 0,50 €, 1 pièce de 0,20 € et 1 pièce de 0,10 €, afin de ne rendre qu'une seule pièce de chaque catégorie. Cela permet d'éviter l'épuisement d'une catégorie de pièces.

**IMPORTANT : le monnayeur ne rend la monnaie que si l'utilisateur introduit des pièces de 0,50 € ou de 1 €.** Si l'utilisateur introduit des pièces de 0,10 € ou de 0,20 € il doit faire l'appoint.

De plus, le monnayeur **restitue toute la monnaie introduite** dans les 3 cas suivants :

- \* si on introduit des pièces dans l'appareil sans demander de boisson
- \* si on demande une boisson dont le prix est supérieur au total des pièces introduites
- \* si on demande les 2 boissons simultanément [appuie sur les 2 boutons en même temps]

### I - 2 - Définition des variables logiques utilisées

Variables logiques d'entrée		Variables logiques de sortie	
Nom	Événement correspondant	Nom	Action correspondante
<b>M</b>	« bouton <b>MENTHE</b> enfoncé »	<b>R10</b>	« rendre une pièce de 0,10 € »
<b>C</b>	« bouton <b>CASSIS</b> enfoncé »	<b>R20</b>	« rendre une pièce de 0,20 € »
<b>P50</b>	« pièce de 0,50 € introduite »	<b>R50</b>	« rendre une pièce de 0,50 € »
<b>P1</b>	« pièce de 1 € introduite »	<b>R1</b>	« rendre une pièce de 1 € »

## II - Travail demandé

**II - 1** - Complétez la table de vérité ci-dessous [les valeurs en euros inscrites dans cette table de vérité ne sont données qu'à titre de rappel : reportez-vous à la page 1 pour en connaître leur signification exacte] :

Entrées				Sorties			
0,50 €	1 €	0,20 €	0,50 €	0,10 €	0,20 €	0,50 €	1 €
<b>P50</b>	<b>P1</b>	<b>M</b>	<b>C</b>	<b>R10</b>	<b>R20</b>	<b>R50</b>	<b>R1</b>
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

**II - 2** - En partant de la table de vérité ci-dessus, dégagez les équations logiques des sorties **R10**, **R20**, **R50** et **R1**, puis simplifiez-les au maximum en utilisant la méthode de votre choix.

**II - 3** - Proposez un logigramme de chaque sortie, en employant le minimum de portes logiques, et en utilisant seulement les fonctions logiques disponibles dans la librairie **CMOS 4000 series** de Proteus.

Les portes logiques dont vous avez besoin seront prises dans la librairie **CMOS**. Cette librairie contient une série de circuits intégrés, dont les références commencent toutes par **4xxx**, et réalisant chacun une fonction logique particulière. Voici la liste des circuits intégrés de la série **4000** réalisant les fonctions logiques de base :

Fonction logique →	<b>ET-NON</b>	<b>OU-NON</b>	<b>ET</b>	<b>OU</b>	<b>OU-exclusif</b>	<b>OU-exclusif-NON</b>
Avec 2 entrées →	<b>4011</b>	<b>4001</b>	<b>4081</b>	<b>4071</b>	<b>4070</b>	<b>4077</b>
Avec 3 entrées →	<b>4023</b>	<b>4025</b>	<b>4073</b>			
Avec 4 entrées →	<b>4012</b>	<b>4002</b>	<b>4082</b>	<b>4072</b>		<b>Porte NON 4069</b>

**Rappel** : pour obtenir les symboles logiques européens vous devez utiliser les circuits **4xxx.IEC** dans Proteus.

**II - 4** - En prenant les portes logiques dans la librairie **CMOS**, réalisez la saisie du logigramme de la sortie **R10** du monnayeur du distributeur automatique de boissons [et seulement de la sortie **R10** pour l'instant].

**II - 5** - Pour vérifier ce logigramme, placez un générateur d'état logique **LOGICSTATE** sur chacune des 4 entrées [**P50**, **P1**, **M** et **C**.] et une sonde **LOGICPROBE (BIG)** sur la sortie **R10**. Nommez chacune des entrées et la sortie de votre logigramme en plaçant un **LABEL** sur chacun des fils correspondant.

**IMPORTANT : à ce stade du TP, vous ne savez toujours pas si vos équations logiques sont correctes. C'est justement le but de la validation de vous indiquer si la solution que vous proposez est juste ou fausse.**

**II - 6 -** Vous allez maintenant **valider** votre logigramme, c'est à dire faire la preuve qu'il répond bien au cahier des charges demandé. Pour cela, lancez la simulation sous Proteus [touche F12] puis testez la sortie logique **R10** pour chacune des 16 lignes de la table de vérité de la page 2 en modifiant l'état des différentes entrées. Si l'état logique de **R10** est correct pour les 16 lignes de la table de vérité, alors votre logigramme est **validé** et votre équation de **R10** est juste. Si l'état de **R10** est faux sur certaines lignes, revoyez votre équation logique, corrigez votre logigramme, puis recommencez la validation **à partir du début de la table de vérité** jusqu'à obtenir un logigramme parfaitement fonctionnel.

**II - 7 -** De la même manière que précédemment, validez séparément (dans un nouveau projet Proteus pour chacun des logigrammes) les équations des sorties **R20**, **R50** et **R1** du monnayeur.

**II - 8 -** En raison de la hausse des matières premières, le fournisseur des boissons doit augmenter les prix demandés à l'utilisateur sur le distributeur. Chacune des boissons **augmente de 0.20 €**: la menthe à l'eau coûte donc maintenant **0.40 €** et le cassis à l'eau coûte désormais **0.70 €**. Quelles sont alors les nouvelles équations logiques de chacune des sorties **R10**, **R20**, **R50** et **R1** du monnayeur ? Validez votre solution sous Proteus en saisissant le logigramme complet (les 4 équations à la fois) du monnayeur dans un seul projet en utilisant un nombre minimal de portes logiques.

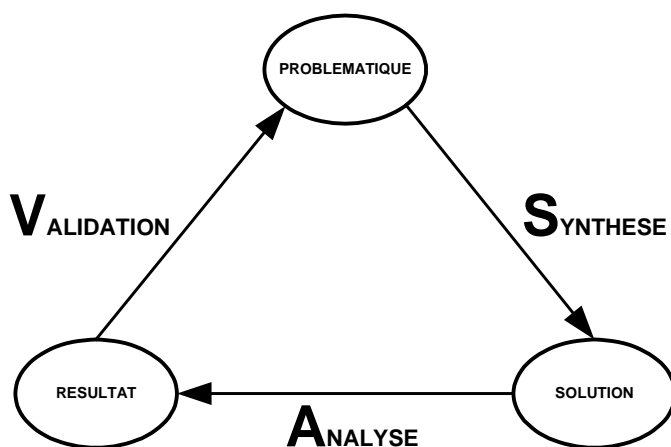
**II - 9 -** Même question avec une nouvelle augmentation de **0.10 €**: la menthe à l'eau est à **0.50 €** et le cassis à l'eau coûte **0.80 €**.

**A RETENIR : « valider » un TP consiste à faire la preuve que la solution que vous proposez est correcte. Cette validation se fait :**

- \* **En autonomie : vous devez conclure seul que votre solution est juste, sans l'aide du professeur**
- \* **En utilisant un simulateur sur ordinateur pour soumettre votre solution aux contraintes réelles (états logiques pour un logigramme, signaux électriques pour un montage analogique, etc.)**
- \* **Si vous constatez que votre solution est fausse il faut recommencer son étude, puis re-valider votre nouvelle proposition**

Le triangle « S.A.V. » retrace cette démarche :

Exemple pour ce TP :



**Triangle S.A.V.**

**Problématique** = trouver le logigramme du monnayeur du distributeur de boissons

**Synthèse** = synthèse de la solution = analysez le problème et construire la table de vérité

**Solution** = Solution proposée pour répondre au cahier des charges = les équations logiques de chacune des 4 sorties = un logigramme

**Analyse** = analyse de la solution proposée = test du logigramme sous Proteus

**Résultat** = résultats donnés par le simulateur = les états logiques des sorties

**Validation** = comparaison des résultats du simulateur avec la problématique initiale

**Le TP est validé si les résultats donnés par l'analyse correspondent à la problématique initiale**