



## TP 1 DISPOSITIF DE SURVEILLANCE DE PHARES



### A. OBJECTIFS

**O5 -Utiliser un modèle de comportement pour prédire un fonctionnement**  
**O6 -Communiquer une idée, un principe ou une solution technique**

- Etablir le logigramme réalisant FP1 selon le fonctionnement décrit.

**Prérequis Notion de table de vérité, logigramme et d'algorithme**

**Utilisation des logiciels ISIS et Flowcode**

**Ressource Logiciels Isis, Flowcode**  
**Fiches Nathan : traitement combinatoire de l'information**

### B. PRÉSENTATION

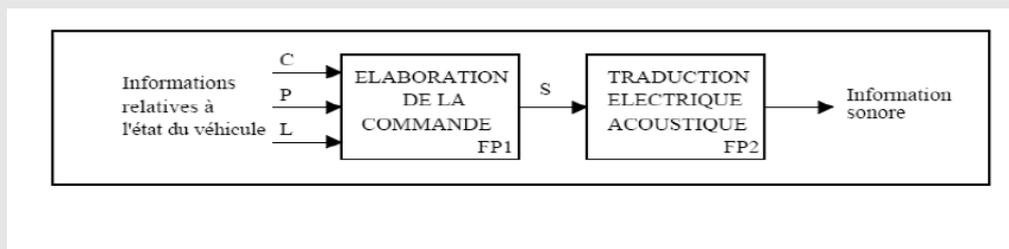
Il est fréquent qu'un automobiliste, ayant quitté son véhicule sans avoir éteint ses feux, le retrouve le lendemain hors d'usage (la batterie s'étant déchargée). Pour éviter ce genre de situation, il peut s'avérer utile d'utiliser un dispositif signalant l'oubli au conducteur distrait.

#### I. ANALYSE FONCTIONNELLE

##### 1. FONCTION D'USAGE

Ce dispositif doit indiquer, par un signal sonore, que les phares sont restés allumés

##### 2. SCHÉMA FONCTIONNEL DE 1ER DEGRÉ



##### 3. FONCTIONNEMENT

L'automobiliste, lorsqu'il a coupé le contact et s'apprête à quitter sa voiture, doit être averti par un signal sonore qu'il a oublié d'éteindre ses phares. Il doit pouvoir laisser délibérément ses phares allumées (en cas de stationnement de courte durée avec mauvaise visibilité par exemple) : le signal sonore doit s'interrompre lorsque l'automobiliste a quitté sa voiture.

Trois variables d'entrée sont donc nécessaires :

- Phares (l) : allumés = 1, éteints = 0.
- Contact (c) : mis = 1, coupé = 0.
- Portière (p) : ouverte = 1, fermée = 0.



Le signal sonore sera activé quand  $S = 1$  et inactivé quand  $S = 0$ .

## E. TRAVAIL DEMANDÉ

### Recherche d'un logigramme

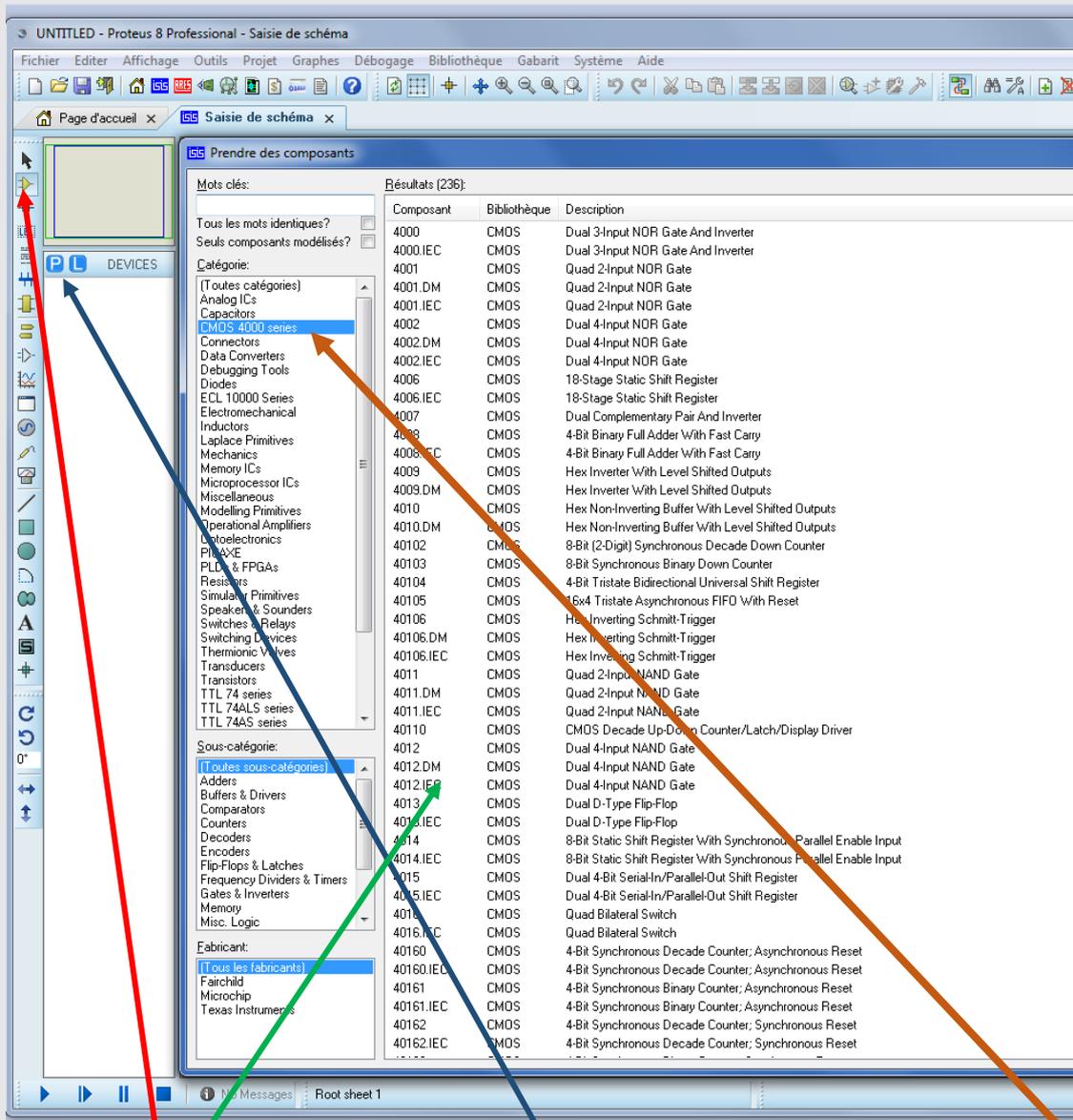
- 1- Etablir la table de vérité du dispositif de commande.
- 2- Déterminer l'équation de S.
- 3- Etablir le logigramme de S à l'aide d'opérateurs CMOS à 2 entrées de votre choix ( série 4xxx).
- 4- Simuler votre solution sous ISIS pour valider son fonctionnement.



Cliquer sur saisie de schéma.

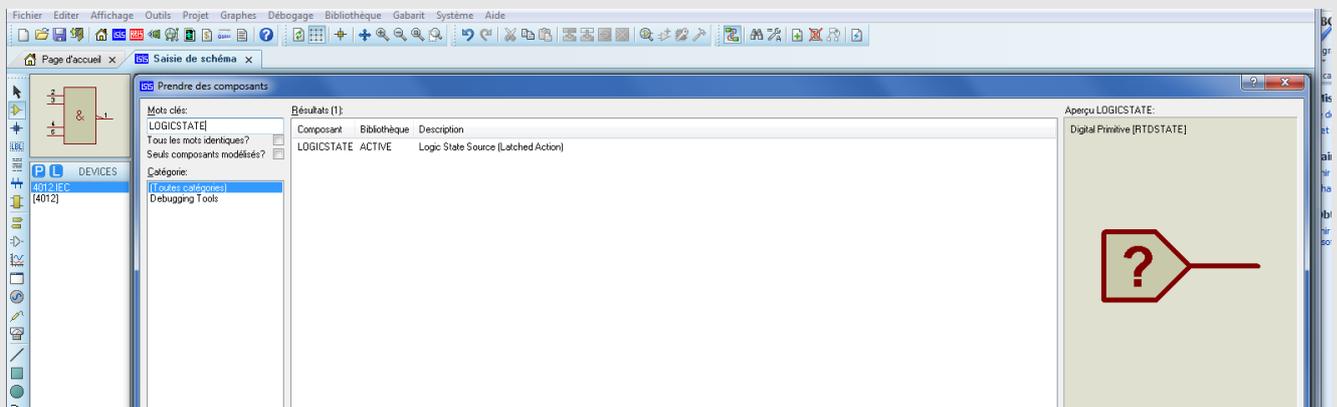
Les portes logiques sont dans la bibliothèque CMOS 4000 série.



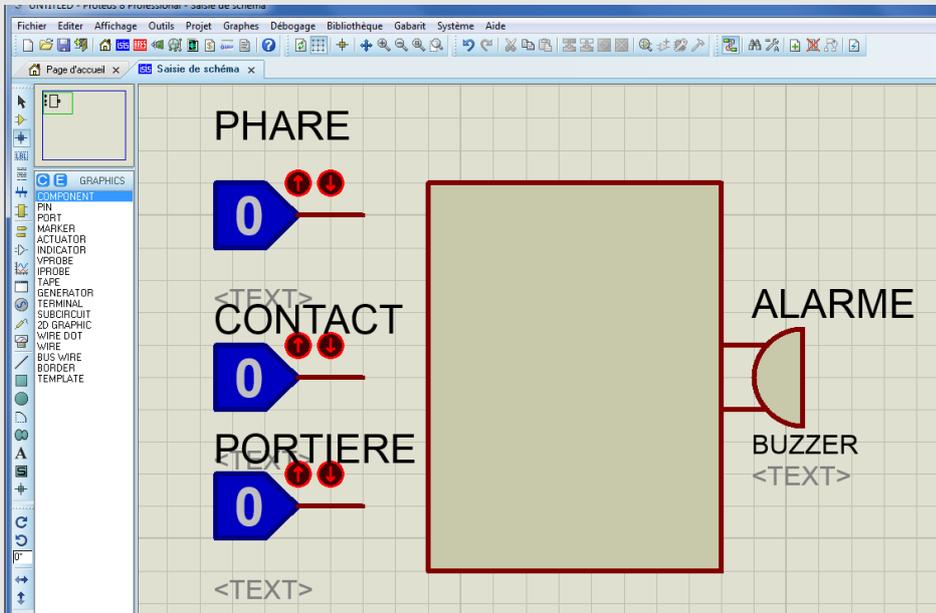
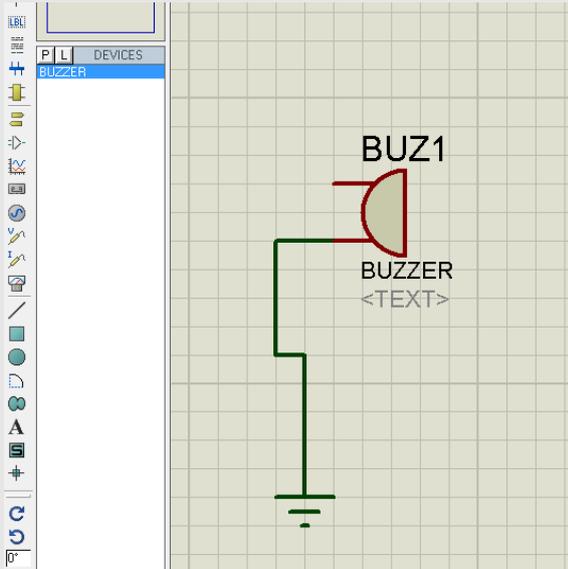


Cliquer sur **mode composant** puis cliquez sur **P** puis choisir vos portes **logique** dans **CMOS 4000** série, **4xxx.iec** (norme européenne).

Trois générateurs logiques LOGICSTATE en entrée Phares (l) , contact (c), portière (p) .



Utiliser le composant buzzer pour simuler l'alarme.



Simuler votre solution sous ISIS pour valider son fonctionnement.

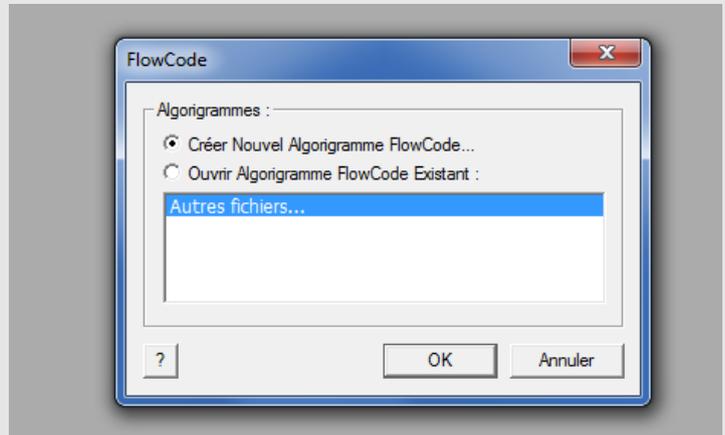
5- Etablir le logigramme de S à l'aide de porte logique Non-ET.  
Simuler votre solution sous ISIS pour valider son fonctionnement.



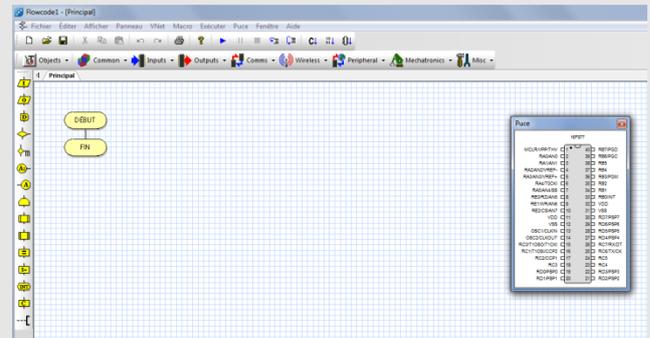
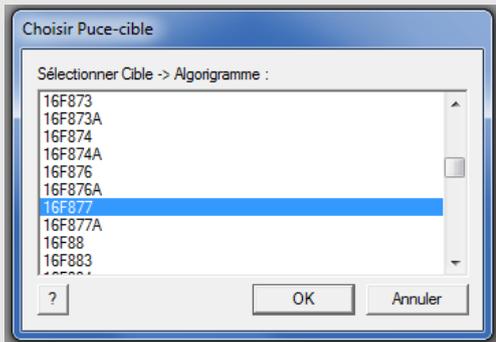
## Recherche d'un algorithme

Nous reprenons le problème mais ici nous utilisons un microcontrôleur pour le résoudre. Voir les affectations des capteurs délivrant les bits l,c, p et S sur la page suivante. Nous utiliserons la simulation sous Flowcode pour le tester et exploitons sa notice si nécessaire.

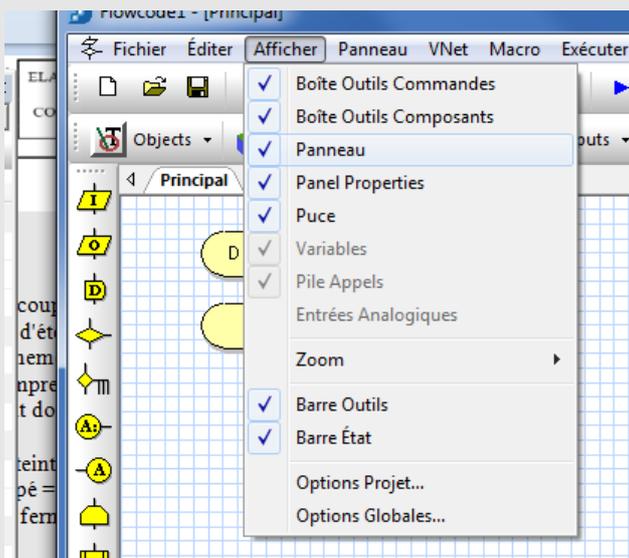
- 1- Lancer le logiciel Flowcode .
- 2- Créer un nouveau Algorithme..



- 3- Choisir la cible 16F887

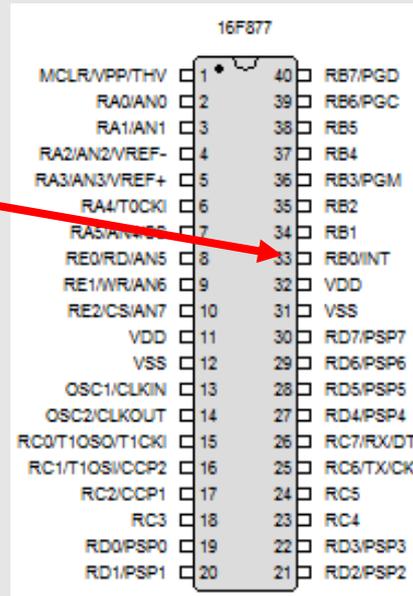


- 4- Afficher le panneau flowcode



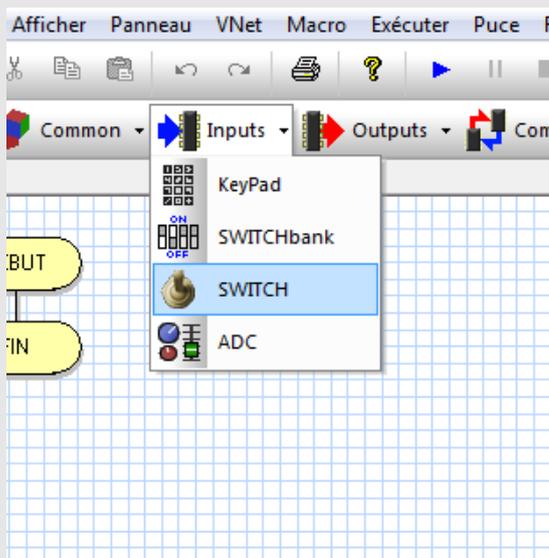
5- Affectation des entrées sorties.

- Phares (l) : RB0
- Contact (c) : RB1
- Portière (p) : RB2
- Le signal sonore S : RB3

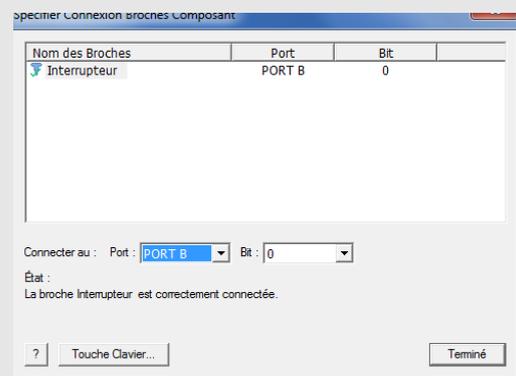
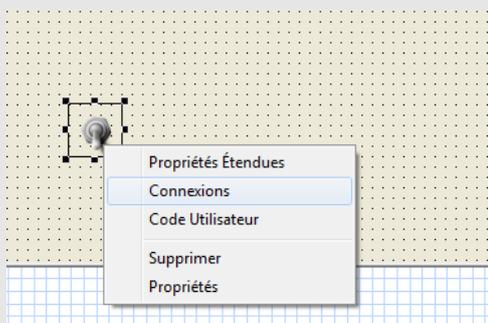


16F877			
MCLR/VPP/THV	1	40	RB7/PGD
RA0/AN0	2	39	RB6/PGC
RA1/AN1	3	38	RB5
RA2/AN2/VREF-	4	37	RB4
RA3/AN3/VREF+	5	36	RB3/PGM
RA4/T0CKI	6	35	RB2
RA5/AN5	7	34	RB1
RE0/RD/AN5	8	33	RB0/INT
RE1/WR/AN6	9	32	VDD
RE2/CS/AN7	10	31	VSS
VDD	11	30	RD7/PSP7
VSS	12	29	RD6/PSP6
OSC1/CLKIN	13	28	RD5/PSP5
OSC2/CLKOUT	14	27	RD4/PSP4
RC0/T1OSO/T1CKI	15	26	RC7/RX/DT
RC1/T1OSI/CCP2	16	25	RC6/TX/CK
RC2/CCP1	17	24	RC5
RC3	18	23	RC4
RD0/PSP0	19	22	RD3/PSP3
RD1/PSP1	20	21	RD2/PSP2

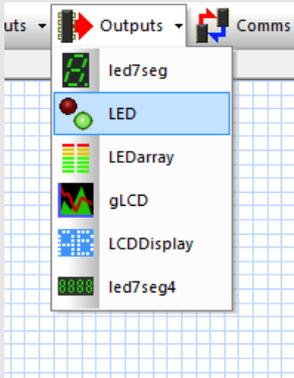
6- Simuler sur le panneau de contrôle de flowcode les entrées et sortie.



7- Puis connecter les switches au pic 16F877.



8- Pour le signal sonore on le simulera avec une Led.

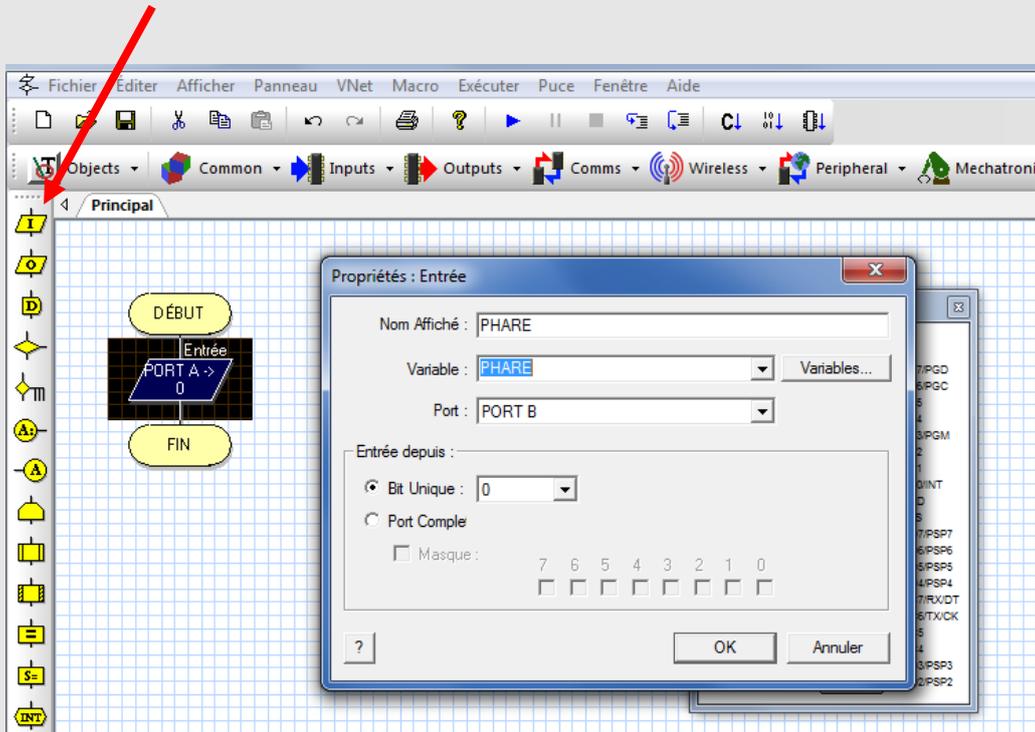


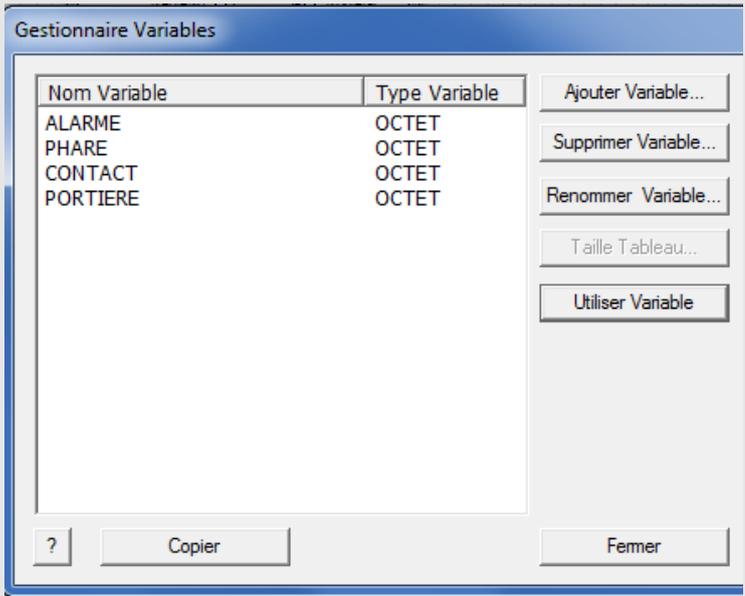
9- Vous devez avoir ceci.



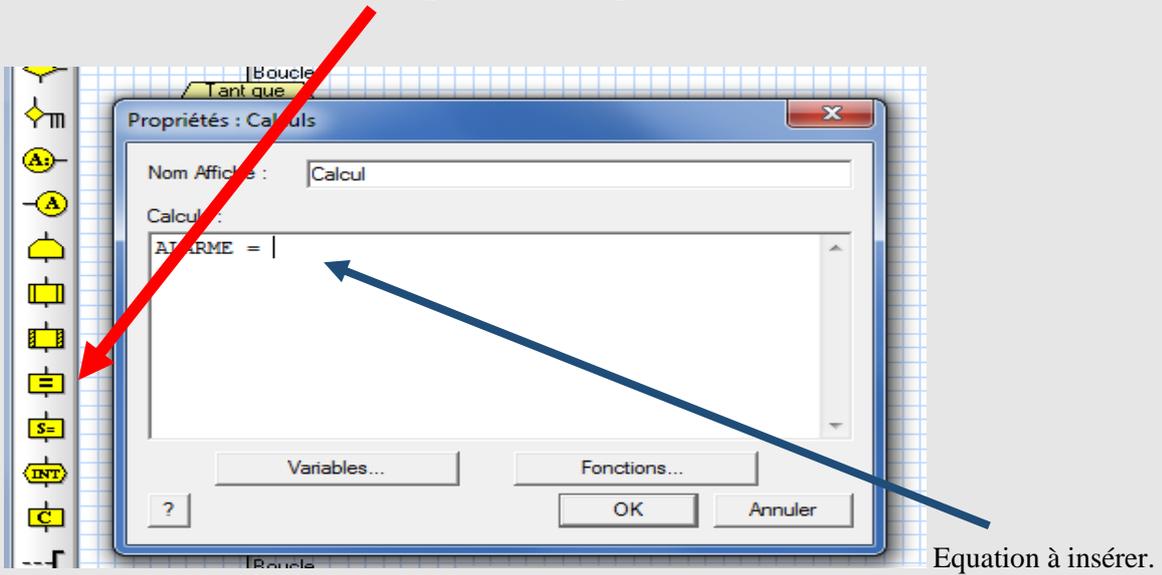
10- Début de l'algorithme.

Tester les Entrées et les stocker dans une variables (ici PHARE).



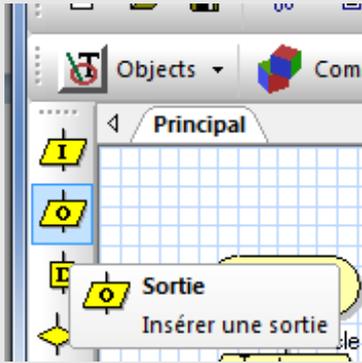


11- Utiliser l'algorithmme calcul pour insérer l'équation de S : alarme.

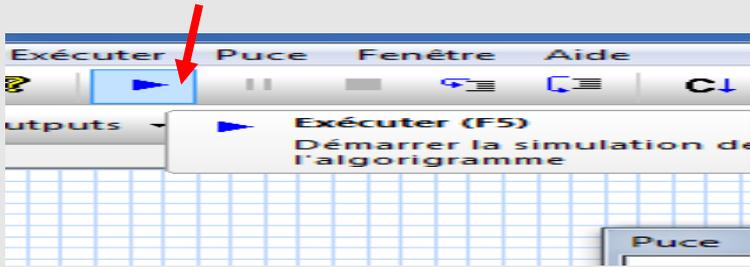


12- Afficher le résultat de la variable de sortie Alarme.



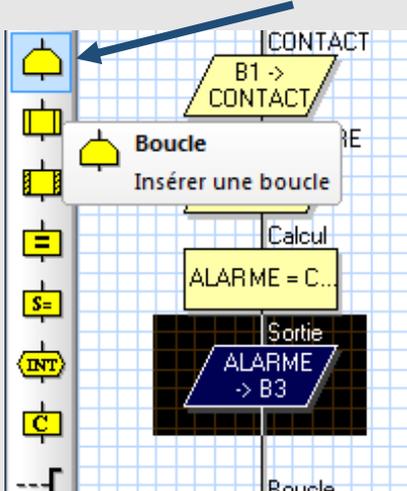


13- Simuler le résultat .



Valider le fonctionnement du système.

14- Utiliser l'algorithme boucle pour un fonctionnement continu.



15- Nous allons simuler les phares par une sortie LED connecté à RB4.  
Réaliser la simulation complète.

