

UTILISATION DE LOGIPIC V2

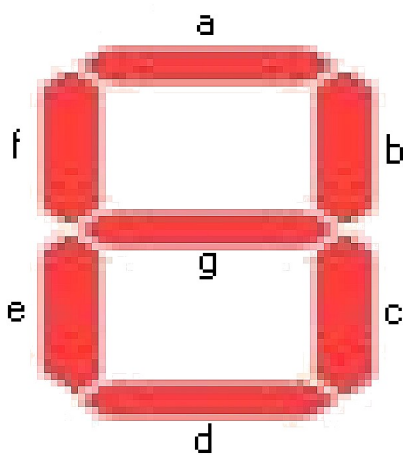
Il s'agit de concevoir et d'écrire des organigrammes permettant la commande de la cible A7S.

1 - FONCTIONNEMENT D'UN AFFICHEUR 7 SEGMENTS

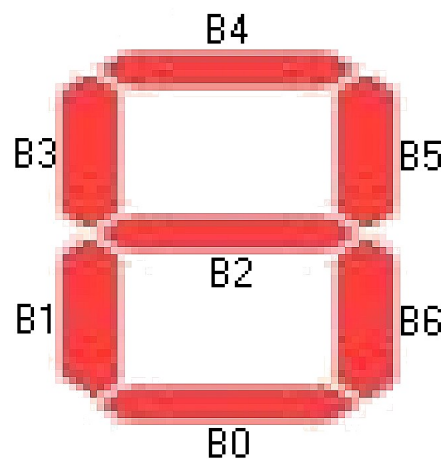
Un afficheur 7 segments permet d'afficher des nombres et certaines lettres de l'alphabet. Son usage est largement répandu dans de nombreux systèmes.

Comme son nom l'indique, l'afficheur est composé de 7 segments qui sont des DEL (Diode Electro-Luminescente) en Anglais on dit LED (Light Emmiting Diode).

Dans un afficheur 7 segments, les segments sont généralement désignés par des lettres de a à g :



L'afficheur que nous utiliserons dispose de 7 segments dont la commande s'effectue avec la logique suivante :



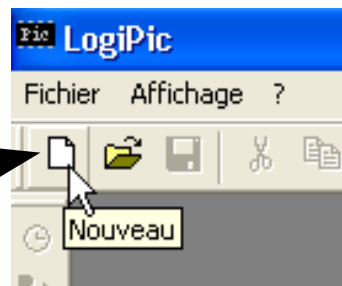
2 - PREMIERE UTILISATION DE LOGIPIC

Il est temps d'écrire notre premier organigramme. Celui ci va permettre de faire clignoter le segment B2 pendant 1 seconde.

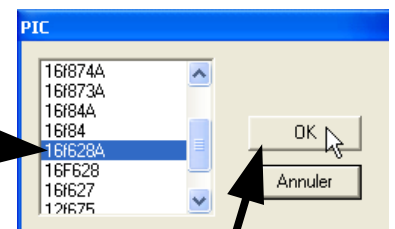
Lancer Logipic V2.05



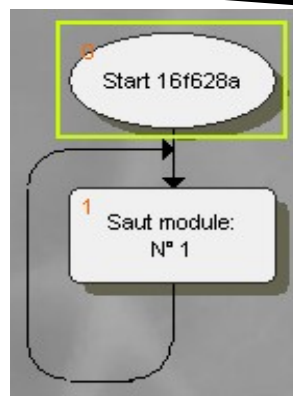
Cliquer sur Nouveau



Sélectionner 16f628A



Le début de l'organigramme se dessine :



Puis OK

N'oubliez pas d'enregistrer votre projet
Aller dans Fichier > Enregistrer sous

nommer votre fichier « prog1 »

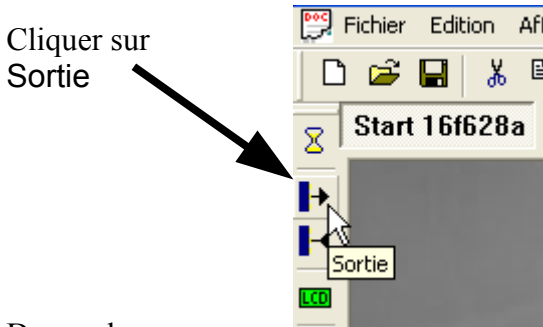
Maintenant que le fichier est correctement positionné, un simple clic sur la disquette permettra d'enregistrer votre travail.



Comme dans toutes les applications Windows, **il faut enregistrer régulièrement** son travail sous peine de tout perdre en cas de problème...

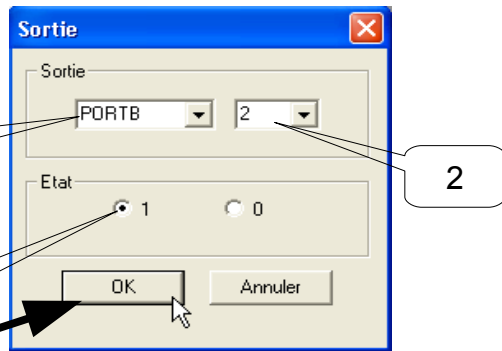


Pour allumer le segment B2, il va falloir préciser que l'on positionne la sortie B2 à 1.



Cliquer sur Sortie

Demander



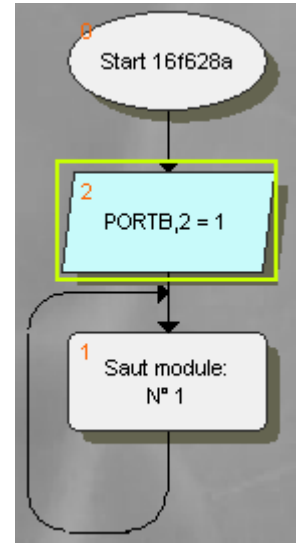
PORTB

2

1

Puis OK

vous obtenez :

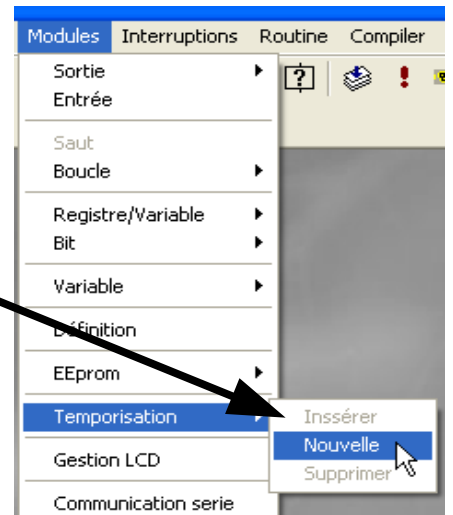


Pour effectuer le clignotement, nous avons besoin d'un délai de 1 seconde pour notre organigramme. Ce délai s'appelle une « Temporisation »

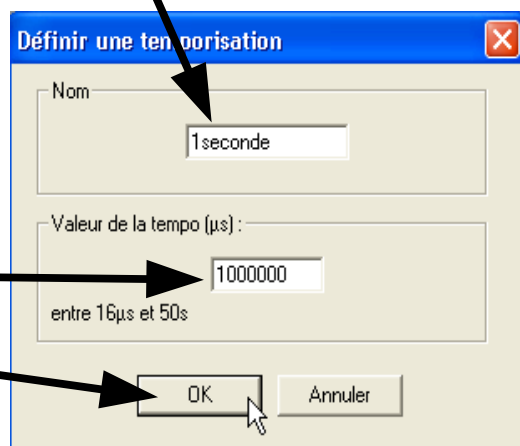
Aller dans Modules > Temporisation > Nouvelle

Il faut indiquer à LogiPic le temps que l'on souhaite. Dans notre cas il s'agit d'une seconde.

Donner un nom, par exemple « 1seconde »



Par défaut, LogiPic positionne une temporisation de 1 seconde soit 1000000µs



Valider par OK



Une action s'appelle un module

En électronique, la seconde est rarement utilisée. On utilise des sous multiples comme la milliseconde (ms) ou la microseconde (μ s).
 C'est comme les distances avec les kilomètres (Km) à la place des mètres (m)...
 1seconde = 1000 ms
 1ms = 1000 μ s
 donc 1000000 μ s = 1seconde

Nous voulons mettre une temporisation à l'issue du positionnement du PORTB,2 = 1
 Vérifier que vous êtes toujours sur PORTB,2 = 1 (il s'entoure en jaune)

Puis sélectionner le sablier qui correspond à la temporisation

Aller chercher la temporisation de 1seconde

Valider avec OK

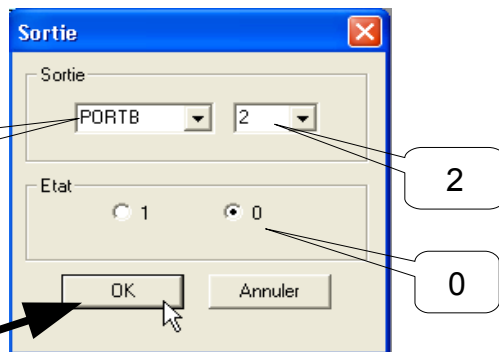
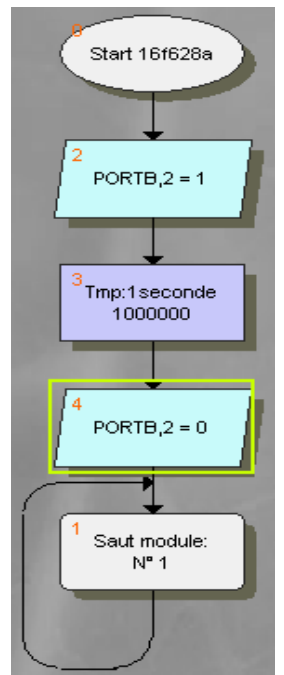
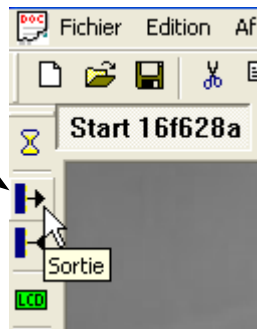
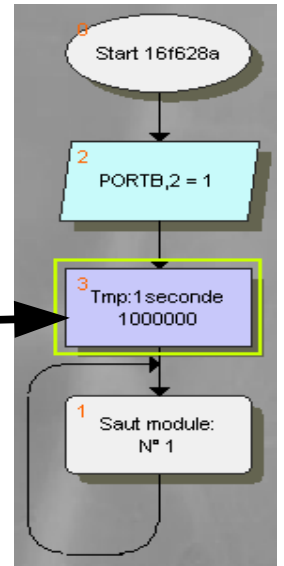
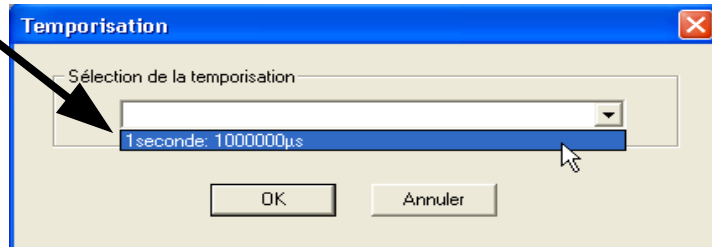
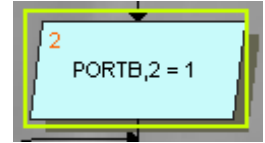
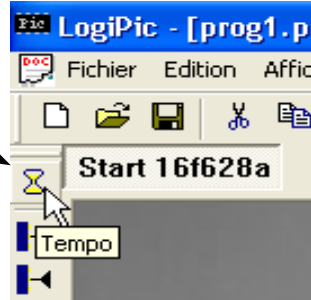
Une temporisation se positionne

Le segment B2 s'allume donc pendant 1seconde, nous allons donc le repositionner à 0, il suffit de positionner le PortB,2 à 0.

Cliquer sur Sortie

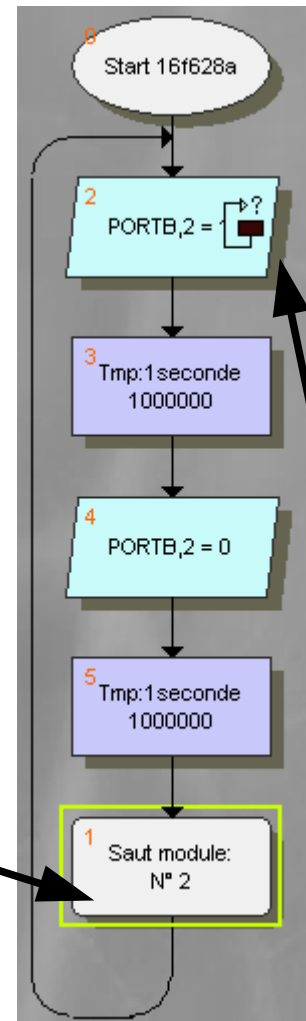
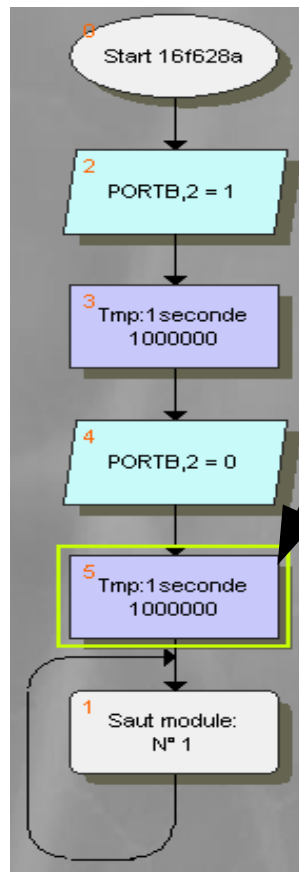
Demander

Puis OK



vous obtenez :

Il va falloir remettre une temporisation pour que le segment s'éteigne pendant une seconde
Sélectionner le sablier et demander la temporisation d'une seconde :



Il manque au programme un rebouclage pour fonctionner de façon clignotante :

Double-cliquer sur
Saut module:
N°2

le curseur se transforme en



Positionner le curseur sur PORTB,2 = 1 pour reboucler l'organigramme

N'oubliez pas d'enregistrer votre programme...

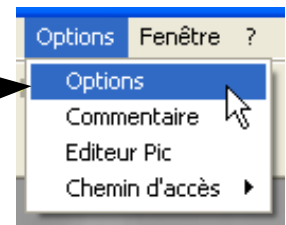
Récapitulons l'organigramme :

Le port B2 est mis à 1 , le segment g s'allume,
l'allumage s'effectue pendant 1 seconde,
Le port B2 est mis à 0 , le segment g s'éteint,
l'extinction s'effectue pendant 1 seconde,
le cycle reprend.

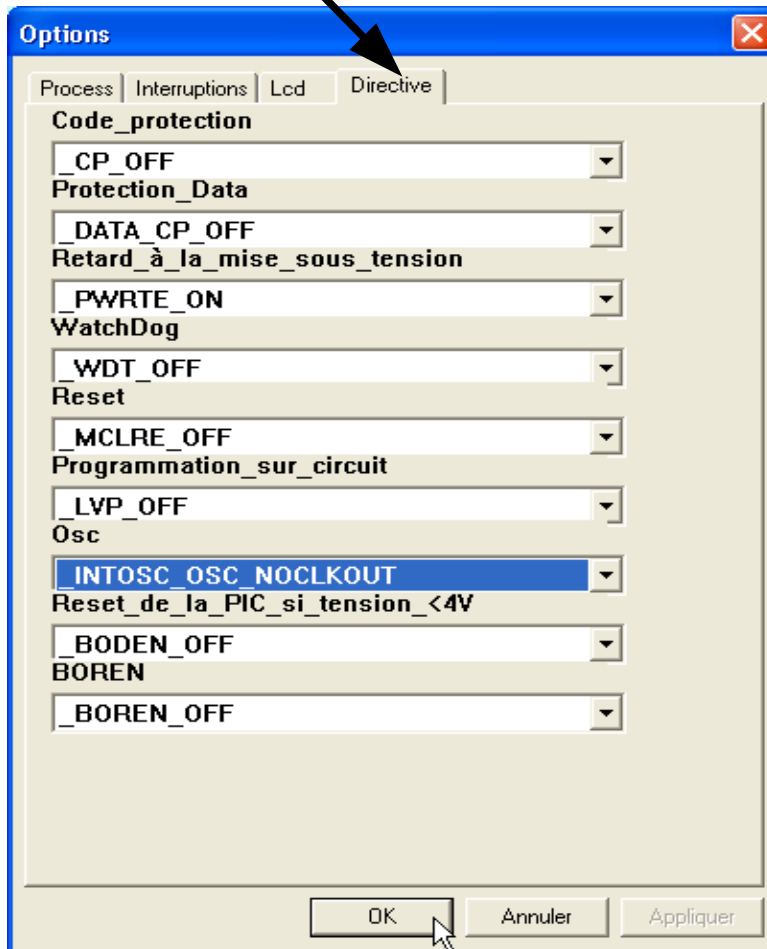
Votre premier programme est terminée, nous allons pouvoir l'essayer....

Avant de tester votre organigramme, il est très important de vérifier les directives.

Aller dans Options > Options



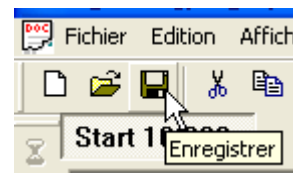
Aller dans l'onglet Directive, positionner les directives comme ci-dessous :



Vérifier bien les directives
sinon, la cible ne
fonctionnera pas...

Valider par OK

N'oubliez pas d'enregistrer votre programme...



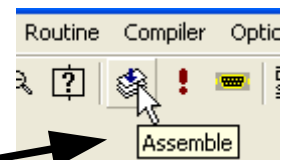
Pour pouvoir envoyer le programme à la cible,
il est indispensable de passer par 3 étapes :

- assemblage,
- compilation,
- transfert vers la cible.

Assemblage :

Il faut transformer l'organigramme dans un langage utilisé par le Pic.
Ce langage s'appelle de « l'assembleur ». On l'appelle aussi « Code source ».

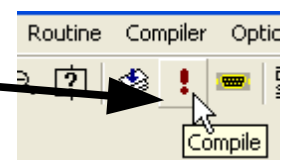
Cette étape s'effectue très simplement en cliquant sur l'icône Assemble



Compilation :

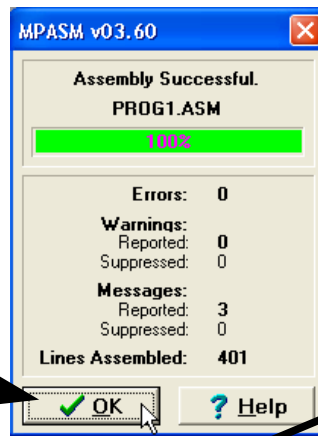
Il faut transformer l'assembleur qui est un langage informatique en données exploitables par le Pic.
Cette opération est réalisé par le logiciel MPASMWIN

Cette étape s'effectue en cliquant sur l'icône Compile

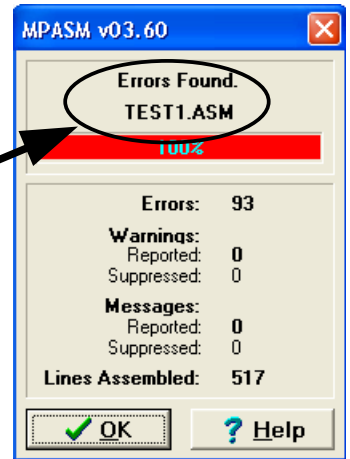


MPASMWIN se met automatiquement en route

Si tout se passe bien vous obtenez l'information suivante :



Fermer MPASMWIN en validant OK

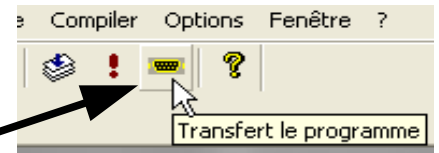


Si l'assemblage indique une barre rouge, il y a des erreurs

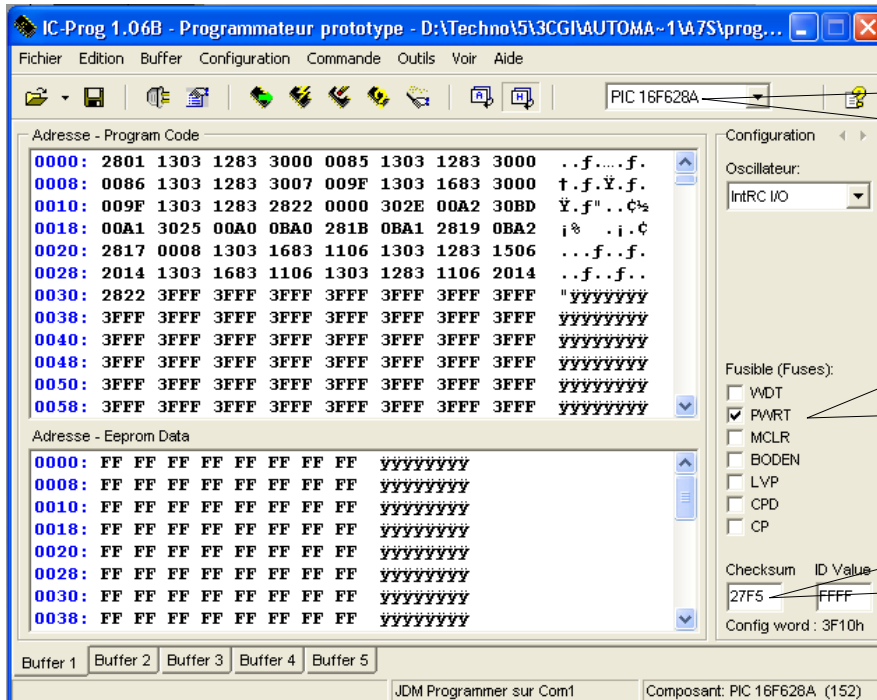
Cela indique l'impossibilité pour MPASMWIN de transformer l'assembleur en données. Il existe d'innombrables raisons de provoquer des erreurs de compilation mais **c'est souvent un nom de variable ou de routine non conforme** (utilisation d'accent ou de signes). Cela peut provenir aussi d'un organigramme dépassant les possibilités du circuit Pic.

Transfert :

Il faut maintenant transférer les données vers la cible. Cette opération est réalisé par le logiciel IC-Prog.



Cette étape s'effectue en cliquant sur l'icône Tranfert IC-Prog se met automatiquement en route



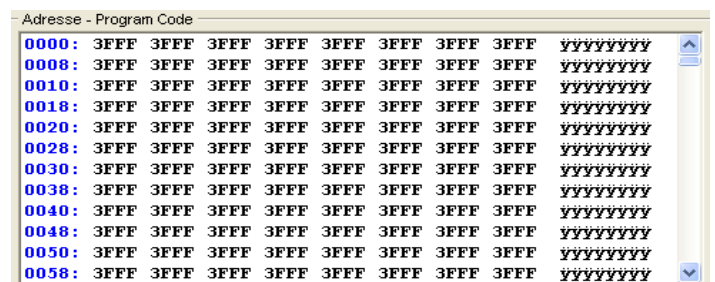
La cible est équipée d'un 16F628A

Si cette configuration n'est pas identique... Les directives de Logicpic sont à vérifier.

Si vous avez effectué le programme comme indiqué. Vous devez obtenir un Cheksum identique : 27F5



Si le Program Code est toujours à 3FFF. Vérifier que votre nom de sauvegarde de projet n'est pas supérieur à 8 caractères....



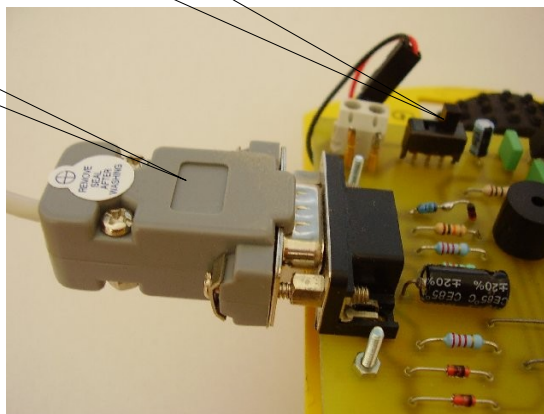
L'envoi du programme est réaliser en cliquant sur l'icône Tout programmer



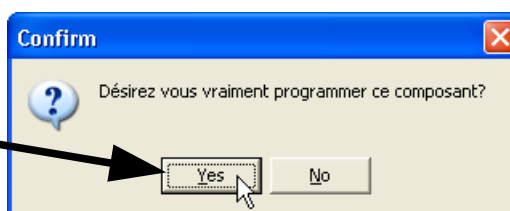
Cable RS232

Mise en route

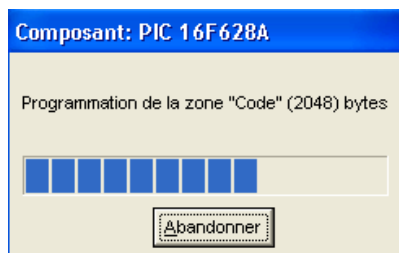
Il est indispensable d'avoir la cible (A7S) reliée par l'intermédiaire du câble RS232 et de mettre en route la cible(diode rouge allumé en face avant).



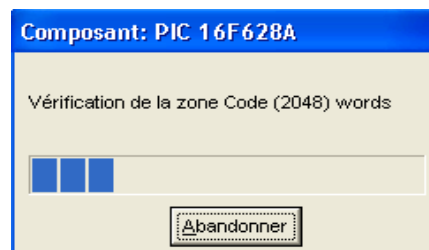
Une demande de confirmation s'affiche : Valider par OK



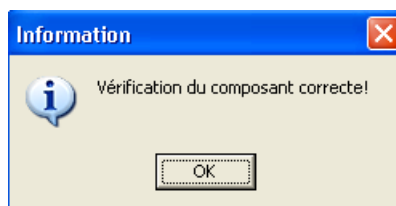
Dans un premier temps Ic-Prog « écrit » dans le Pic :



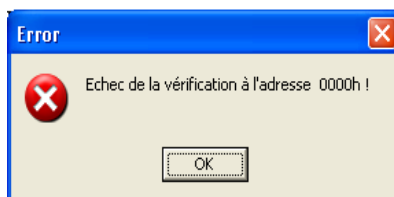
Ensuite la programmation du Pic est vérifiée :



Si la programmation du Pic est correct IC-Prog renvoie l'information suivante :



Si IC-Prog renvoie l'information Error, Vérifier si le câble est bien connecté, si la cible est sous tension et si la configuration d'IC-Prog est conforme.

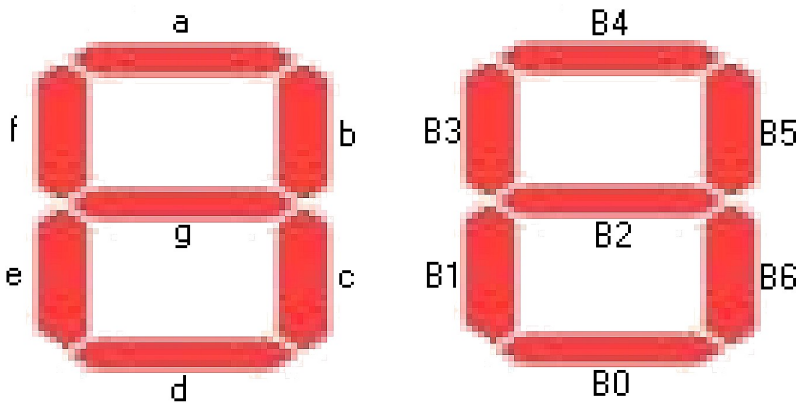


Dans ce cas là, vérifier les options de Logipic mais aussi la configuration Hardware de Ic-Prog. Réessayer de reprogrammer la cible.

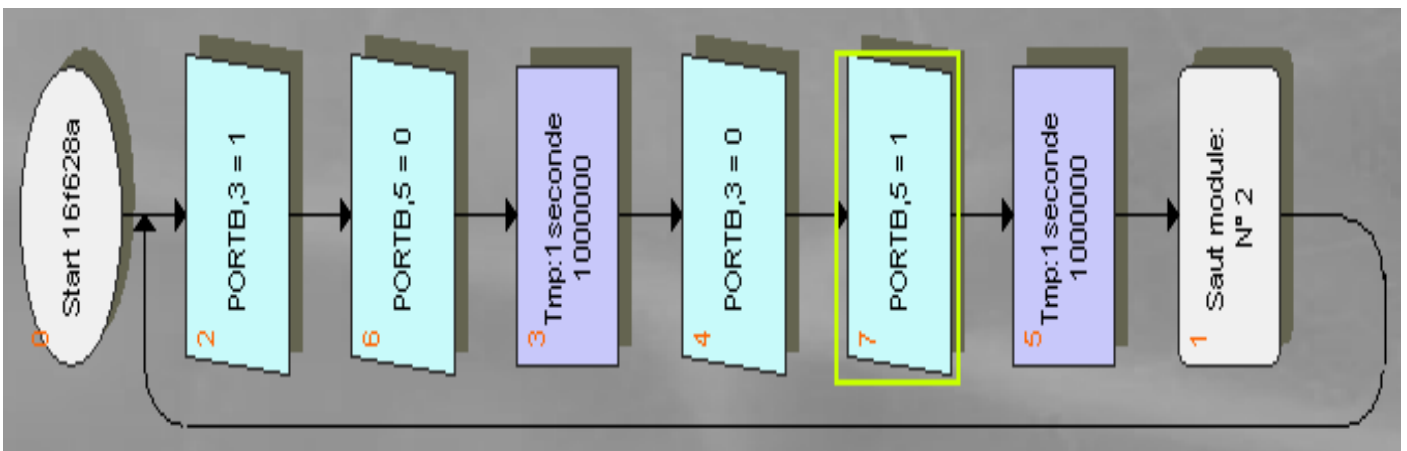
Si la vérification est correcte, A7S se met en action et affiche 0 en clignotant. A7S peut dorénavant fonctionner en « autonome » sans le câble RS232.

3 - TRAVAIL A EFFECTUER

Ecrire un organigramme (appeler le prog2) capable de faire un chenillard (allumage successif) des segments f et b, c'est à dire le PORTB3 et le PORTB5, temporisation de 1seconde.



Je vous propose l'organigramme suivant :



essayer le programme sur la cible.

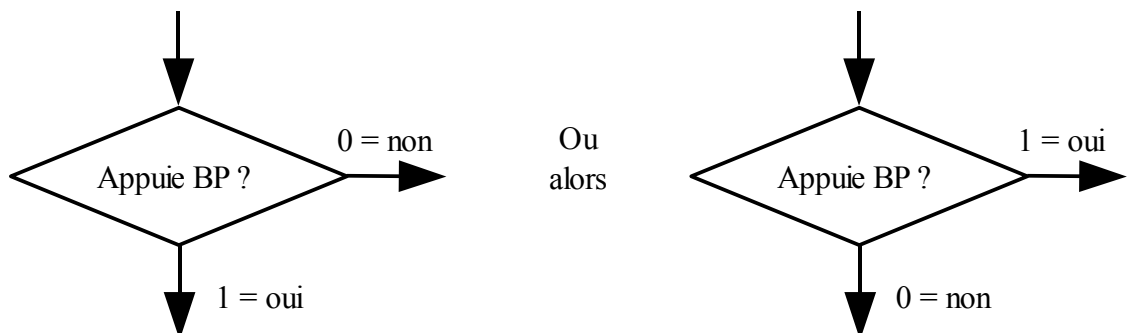
Ecrire un organigramme (appeler le prog3) capable de faire un chenillard (allumage successif) des segments a, b, c, d, e et f , temporisation de 1seconde.

4 - LES TESTS

Si l'on veut utiliser l'information d'un capteur, on utilise un test. Un test dispose toujours de 2 sorties représentatives de l'état du capteur. A7S dispose de 2 capteurs, ce sont 2 boutons-poussoirs (ou BP) dont l'état correspond à la logique suivante :

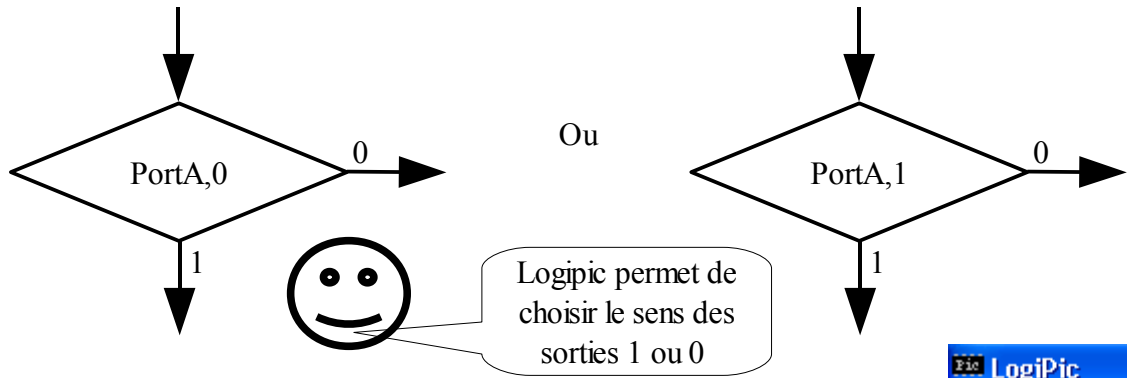
- Bouton-poussoir non actionné fournit un 0,
- Bouton-poussoir actionné (c'est à dire un appui sur le Bouton-Poussoir) fournit un 1.

Le test est symbolisé par un losange dont les sorties sont représentatives de l'état du bouton-poussoir, en fonction de l'organigramme, il est possible de choisir le sens des sorties.

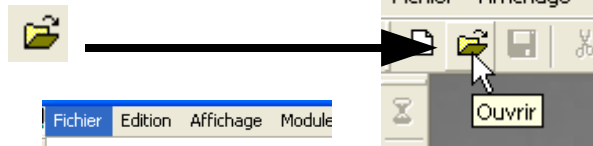


Logipic fonctionne avec des tests sur le portA ou le portB, cependant, le PortB est occupé par l'affichage des segments de l'afficheur.

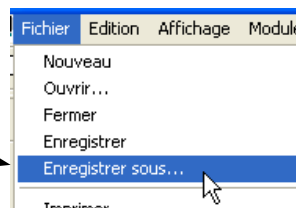
Les 2 boutons-poussoirs d'A7S sont câblés sur le PortA0 et le PortA1. Il est donc possible de tester les boutons-



Ouvrir le projet prog1.prj en utilisant l'icône Ouvrir :

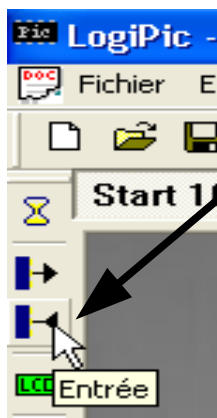


Enregistrer sous le nouveau nom : prog4.prj



Positionnez vous sur la temporisation.

Effectuer un test en utilisant l'icône Entrée:

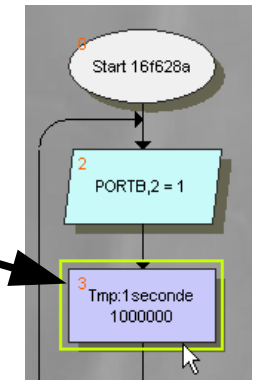


Sélectionner le PORTA,0



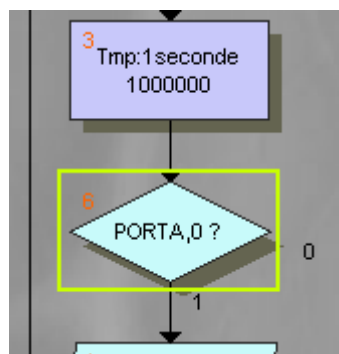
permet de choisir le sens des sorties 1 ou 0

Puis l'entrée 0



Validez par OK

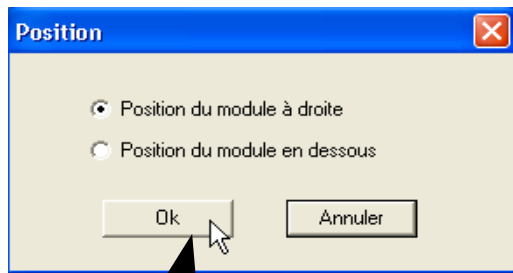
Vous obtenez le test suivant :



Effectuer un saut en utilisant l'icône :



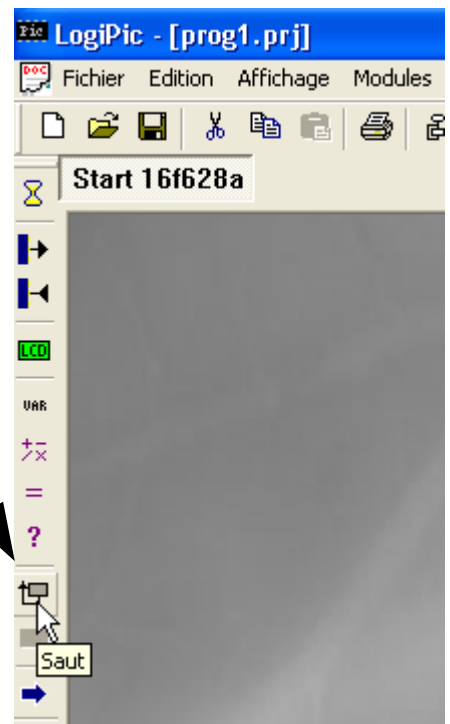
La boîte de dialogue suivante s'affiche :



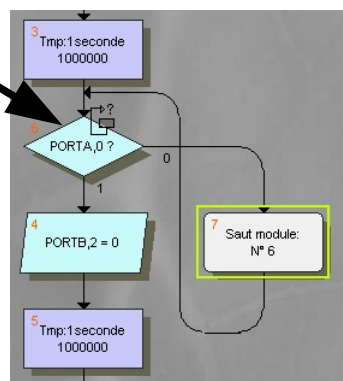
validez par OK



Cela permet de choisir ou mettre le saut



Reboucler le test sur lui-même :
en cliquant sur le module de test

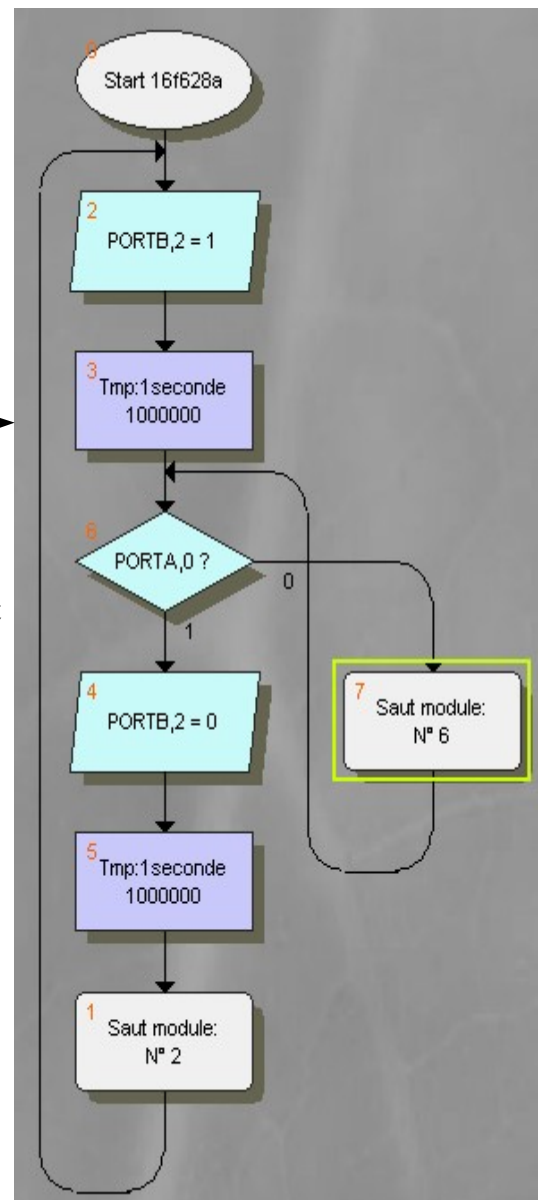


Vous devez obtenir l'organigramme suivant :

Tester l'organigramme en l'envoyant sur A7S (voir page 6).

Le programme permet l'allumage du segment g, dès l'appui du Bouton poussoir « A0 », le programme permet le clignotement du segment g.

A7S (voir page 6).



5 - COMMANDE DU BEEPER

A7S dispose aussi d'un buzzer dont la commande s'effectue en utilisant la sortie A2.
 Nous pouvons donc effectuer un « beep » en effectuant la commande suivante :

Nous allons utiliser le portA,2

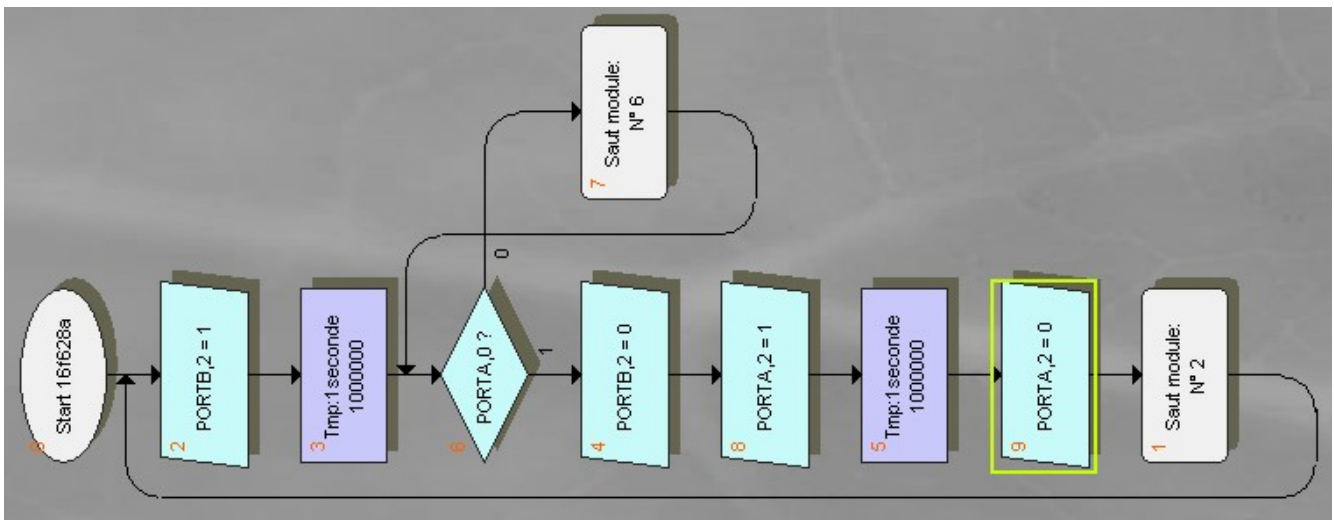


Valider avec OK



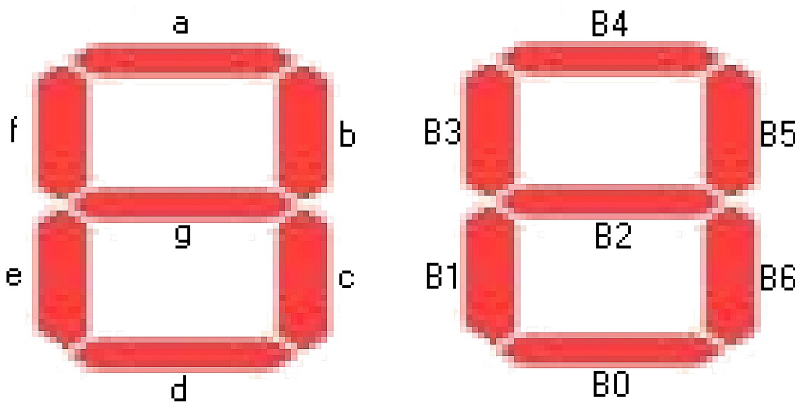
Pour ne pas que le buzzer beep de façon continue, il va falloir mettre une temporisation puis repositionner le PortA2 à 0...

Essayer le programme **prog5.prj** suivant :



6 - TRAVAIL A EFFECTUER

Ecrire un organigramme (appeler le prog6) capable d'afficher le segment g à la mise en route, un 0 dès l'appui du Bouton-poussoir sur le portA,0 et un 1 dès l'appui du Bouton-poussoir sur le portA,1.

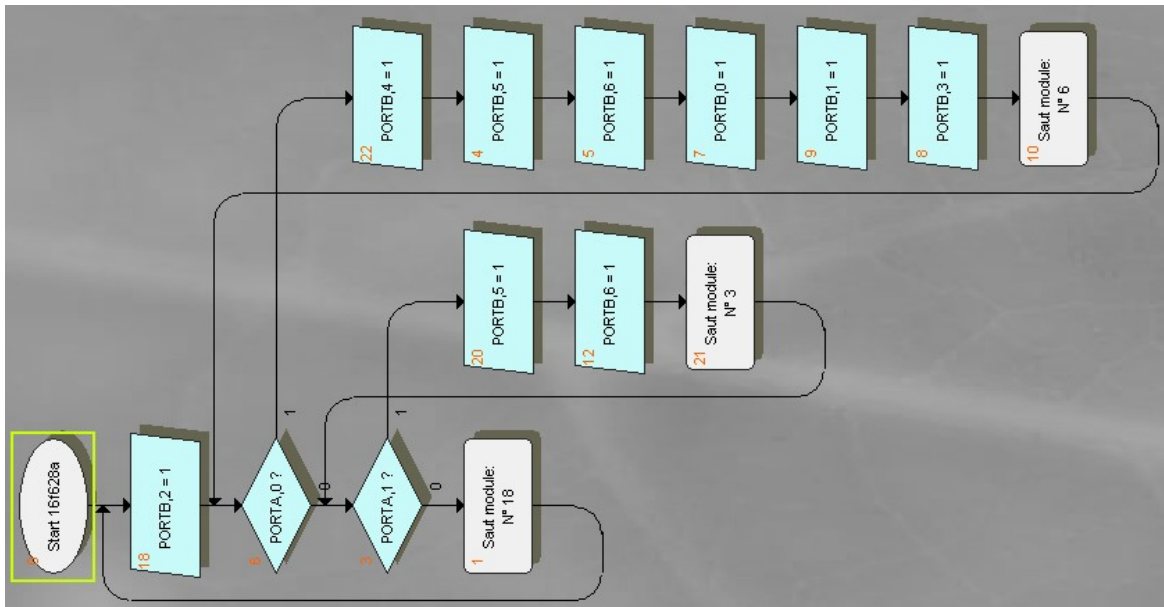


Segment g allumé si portB,2 à 1

affichage 0 si portB,4 à 1
 portB,5 à 1
 portB,6 à 1
 portB,0 à 1
 portB,1 à 1
 portB,3 à 1

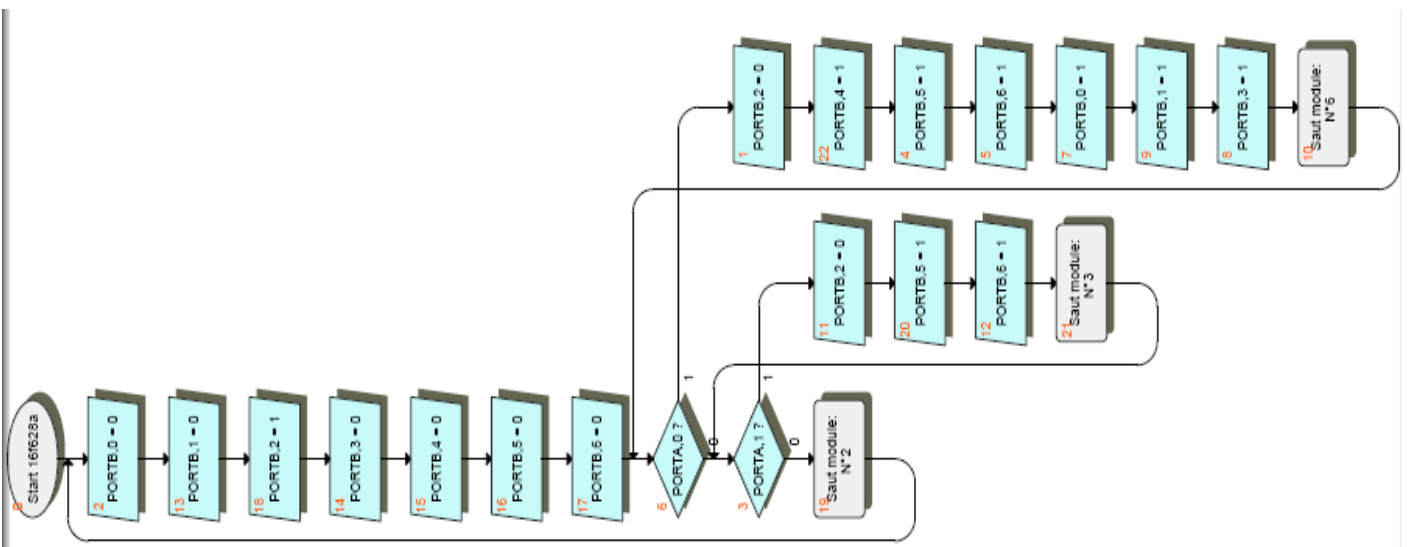
affichage 1 si portB,5 à 1
 portB,6 à 1

La première idée d'organigramme est le suivant :



Malheureusement , ce programme n'est pas complètement au point. En effet, il faut configurer toutes les sorties du portB aussi bien en 1 qu'en 0.

Il est très important de positionner tous les segments à chaque passage. Pour fonctionner, le programme doit configurer tous les changements. Le programme suivant est fonctionnel :



6 - LES SOUS PROGRAMMES OU ROUTINES

Il serait long et fastidieux d'écrire des programmes qui utilisent des actions (logipic nomme ses actions des modules) toujours identiques. En informatique, on utilise très souvent des « mini-programmes » qui sont appelés en fonction des besoins. L'utilisation des sous-programmes permet de simplifier et de limiter la taille du programme final.

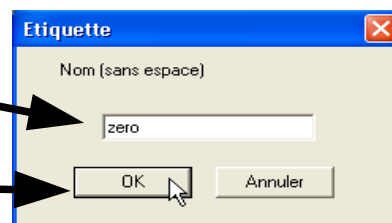
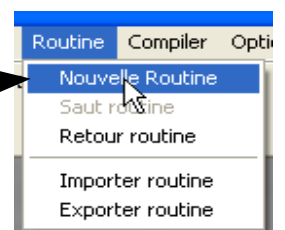
L'exemple suivant est un programme effectuant la même action que précédemment mais il utilise la technique de Sous Programme.

Ouvrir un nouveau programme que vous appellerez prog7 (voir page 2)

Aller dans Routine > Nouvelle Routine

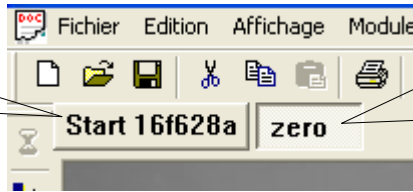
Donner le nom « zero » à ce sous-programme

Valider par OK



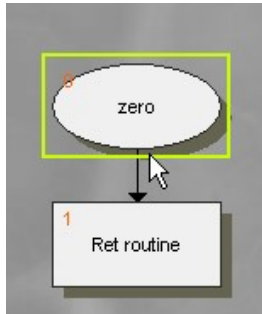
Une nouvelle indication s'affiche :

Cliquer ici pour aller dans le programme principal



Il s'agit du sous-programme ou routine « zero »

Vous pouvez maintenant écrire la routine, positionner le curseur en début de routine

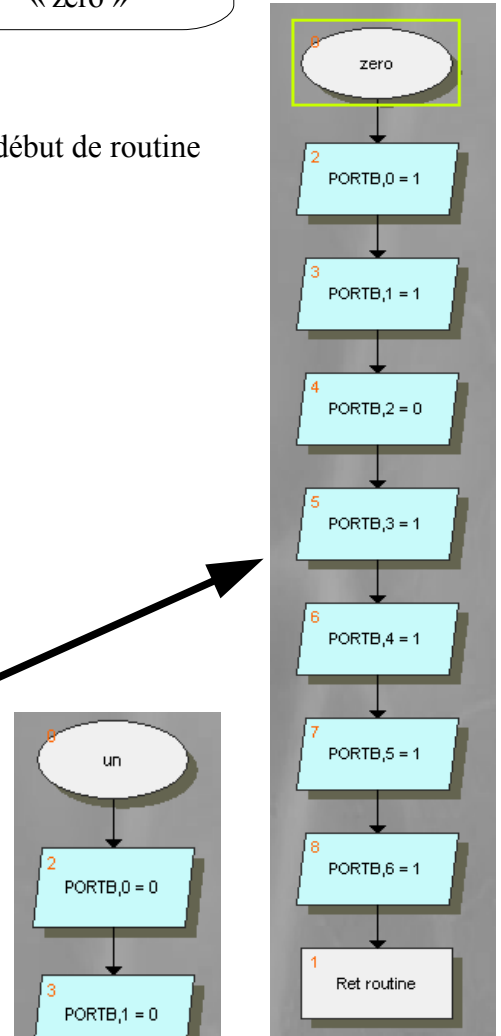


nous voulons afficher un 0,
Pour cela, il suffit de positionner le portB :

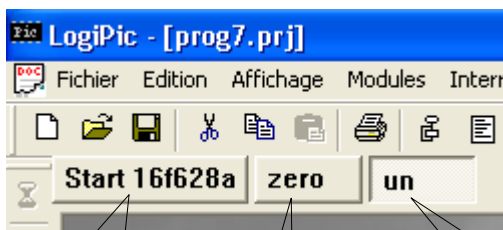
- PortB0 à 1
- PortB1 à 1
- PortB2 à 0
- PortB3 à 1
- PortB4 à 1
- PortB5 à 1
- PortB6 à 1

(Si vous ne savez plus faire : regarder la page 2)

Définir une nouvelle routine qui s'appelle « un »
et qui permet d'afficher un 1



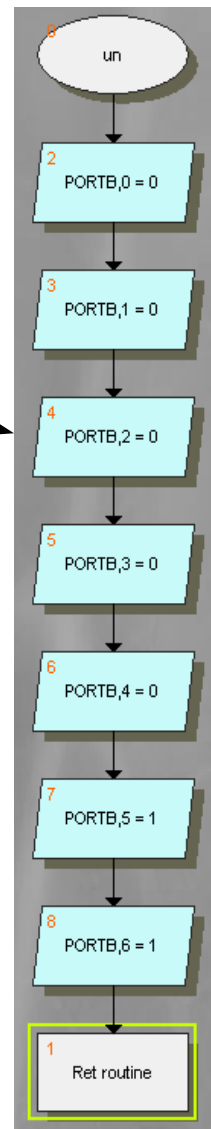
Maintenant que les 2 routines sont écrites
retourner dans le programme principal
en effectuant un double clic sur Start 16f628a




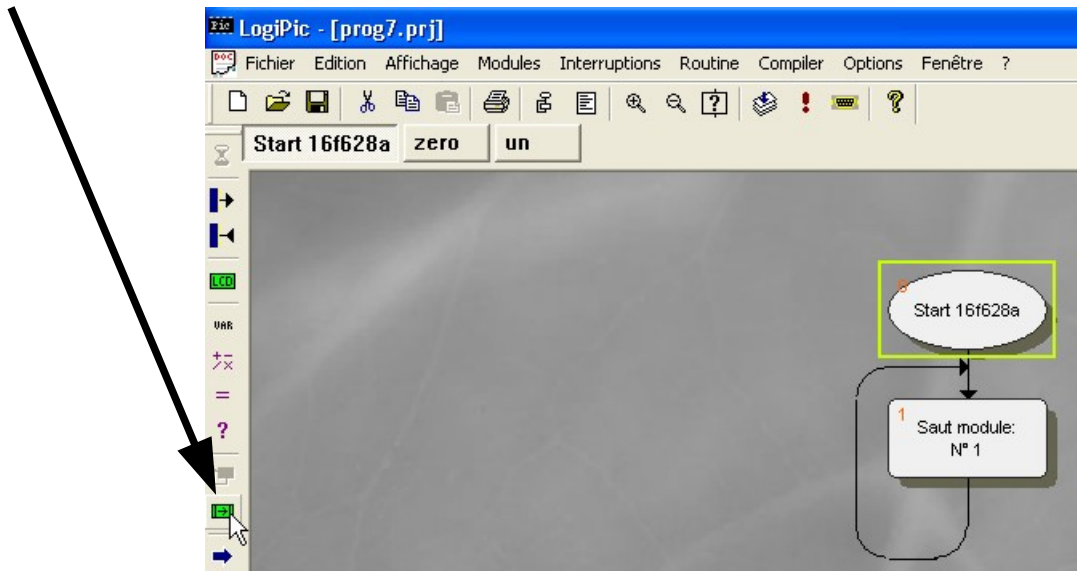
Programme principal

Routine : zero

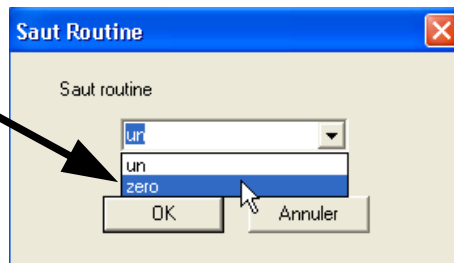
Routine : un



Demander l'appel à la routine en utilisant l'icône 



Demander la routine « zero »



Puis valider par OK

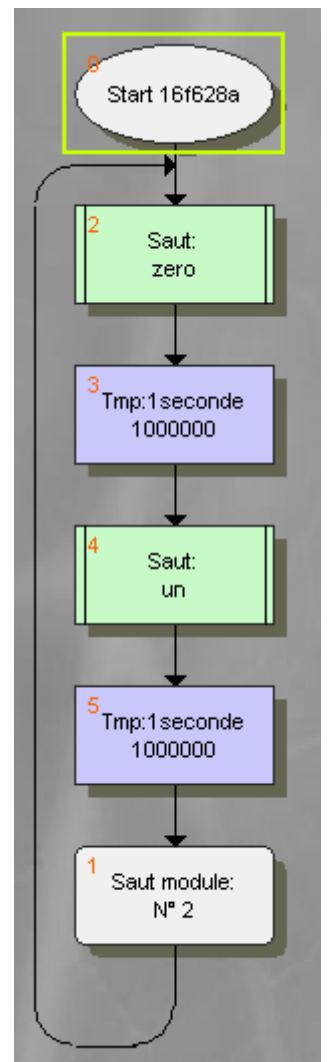
Mettre une temporisation de 1 seconde derrière l'appel de la routine « zero »
(Si vous ne savez plus faire : regarder la page 3)

Demander l'appel à la routine « un » en utilisant l'icône

Mettre une temporisation de 1 seconde

Reboucler le programme sur le module

vous obtenez le programme suivant :



L'utilisation et l'écriture successive de chaque sortie B est assez lourde, il est possible de configurer directement le portB avec la commande `PORTB=B'00000000'` > le B indique une valeur binaire

Il s'agit du bit 7 Il s'agit du bit 0

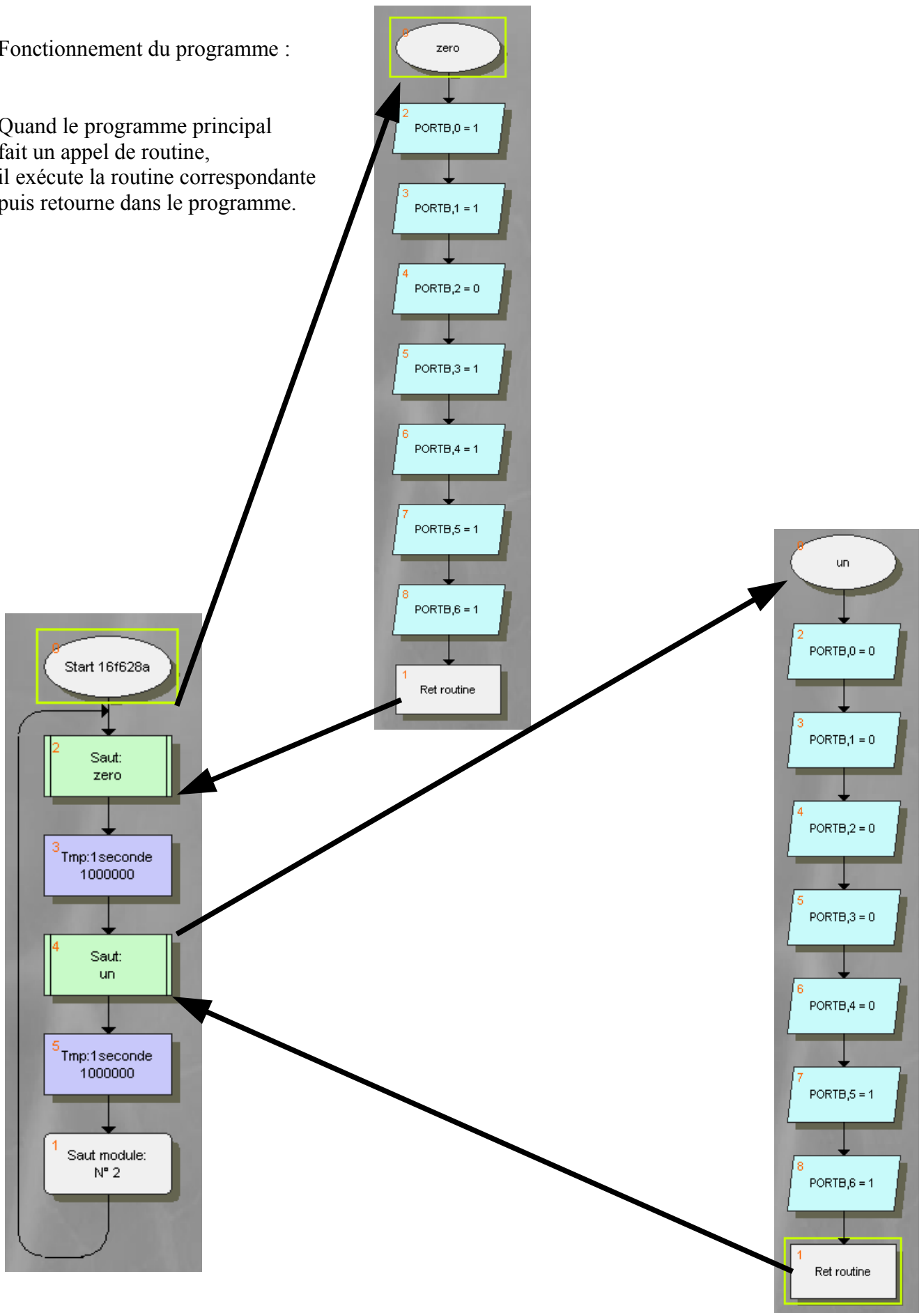
`PORTB=B'01100000'` pour afficher 1
`PORTB=B'01100111'` pour afficher 2

Mais dans ce cas là, il faut configurer le port correspondant en sortie : donc indiquer `TRISB = B'10000000'`

0 indique que le port est en sortie sur le bit correspondant
1 indique que le port est en entrée sur le bit correspondant

Fonctionnement du programme :

Quand le programme principal fait un appel de routine, il exécute la routine correspondante puis retourne dans le programme.



Tester l'organigramme en l'envoyant sur A7S (voir page 6).

8 - EXERCICES

Exercice 1

Ecrire l'organigramme « exo1 » suivant :

Vous devez effectuer le clignotement (allumage et extinction) de B0, B2 et B4 simultanément pendant 1 seconde.

Exercice 2

Ecrire l'organigramme « exo2 » suivant :

Vous devez effectuer un chenillard. C'est à dire l'allumage de B0 puis B1, puis B3 puis B4 puis B5 , puis B6 puis reprendre à B0.

Exercice 3

Ecrire l'organigramme « exo3 » suivant :

vous devez effectuer un Beep lors de l'appuie du bouton poussoir sur A0 et deux Beep lors de l'appuie du bouton poussoir sur A1.

Exercice 4

Ecrire l'organigramme « exo4 » suivant :

En utilisant des sous-programmes (un, deux, trois, quatre, cinq, six, sept, huit, neuf), effectuer un compteur de 1 à 9.

Exercice 5

Ecrire l'organigramme « exo5 » suivant :

A la mise en route, l'afficheur est éteint, dès l'appui du bouton poussoir A0, afficher de façon continu un 0, dès l'appui du bouton poussoir A1, afficher de façon continu un 1. (à voir : utilisation des variables page 18)

Exercice 6

Ecrire l'organigramme « exo6 » suivant :

En partant de « exo5 », mettre en route le compteur lors de l'appuie du bouton poussoir sur A0 et qui s'arrête lors de l'appuie du bouton poussoir sur A1.

Exercice 7

Ecrire l'organigramme « exo7 » suivant :

En partant de « exo4 », mettre en route le compteur lors de l'appuie du bouton poussoir sur A0 et qui s'arrête lors de l'appuie du bouton poussoir sur A1.

9 - UTILISATION DE VARIABLES

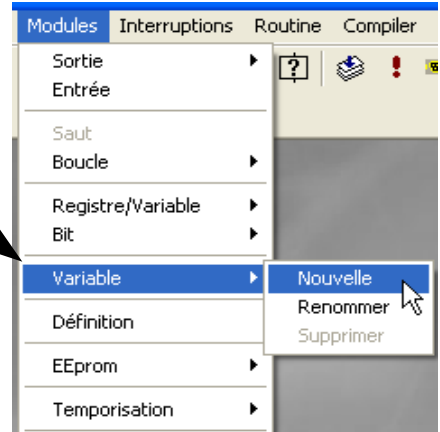
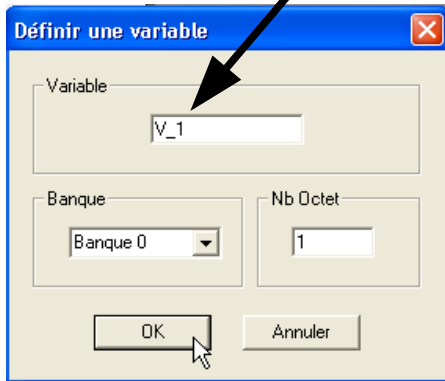
Les variables permettent d'associer un nom à une valeur chiffrée. Logipic utilise généralement des valeurs codées sur 1 octet, cette valeur pourra donc évoluer entre 0 et FF en base hexadécimale ou 0 à 255 en base décimale.

Le programme suivant permet de mémoriser l'appui sur les boutons poussoir et utilise les nombreuses possibilités du logiciel Logipic.

Ouvrir un nouveau projet que vous appellerez prog8 en utilisant le circuit 16f628a.

Aller dans Modules > Variable > Nouvelle

Indiquer comme nom de variable V_1

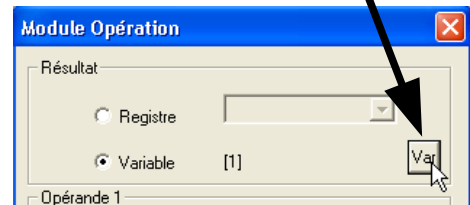


Comme souvent dans un programme, il est souvent indispensable de configurer les variables. Nous voulons positionner la variable V_1 à 0 (donc V_1 = 0)

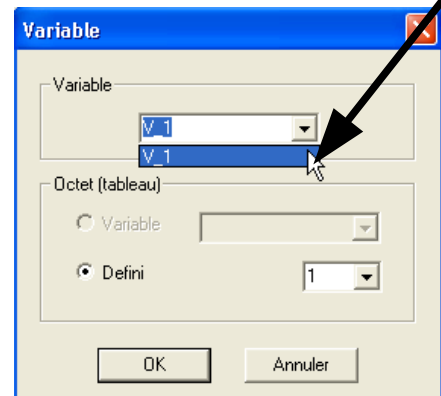
Cliquer sur l'icône =



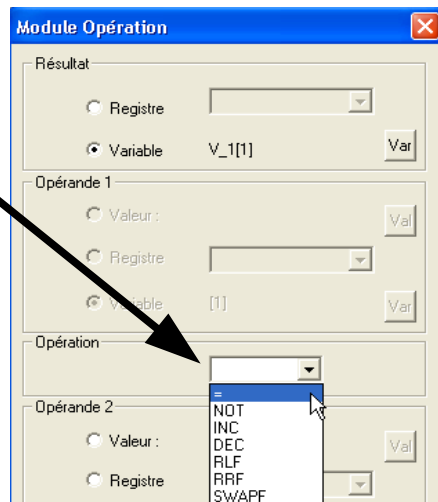
Cliquer sur Var

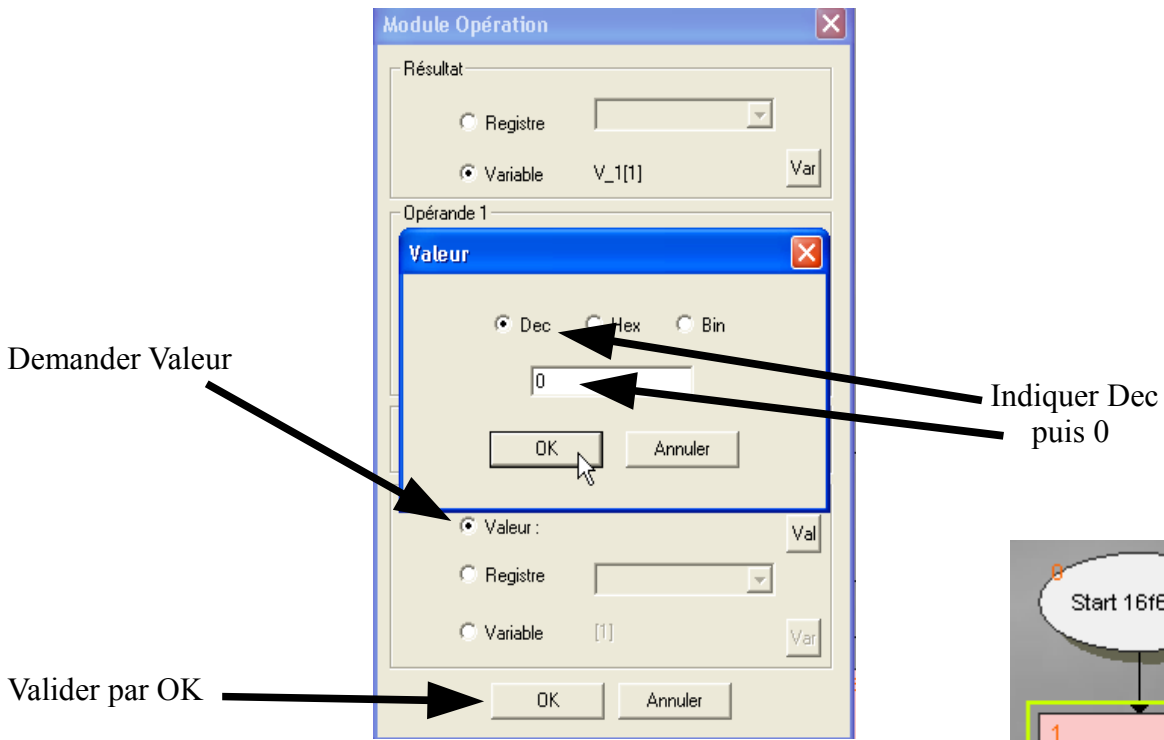


Demander la variable V_1

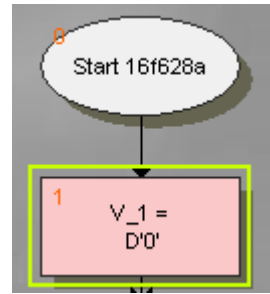


Demander Opération =



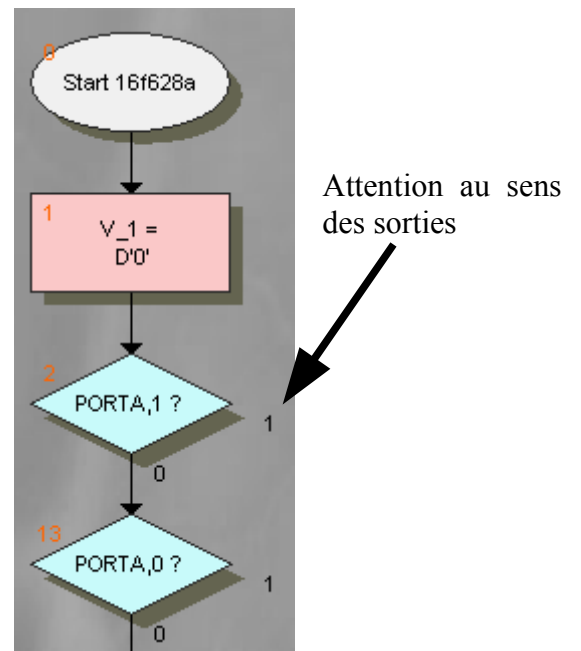


Le module suivant s'affiche :



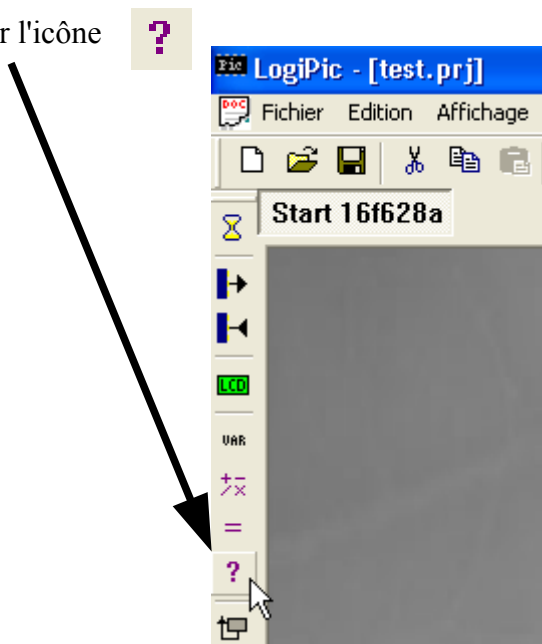
Ce module se comprend simplement : la variable V_1 prend la valeur 0 en décimale.

Positionner deux tests sur les Boutons poussoirs (voir page 9)

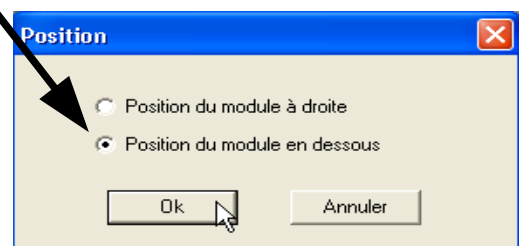


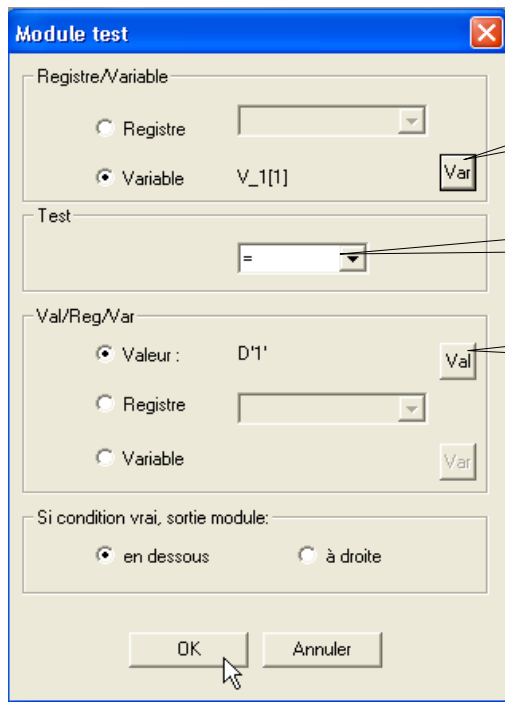
Pour tester la variable V_1

Cliquer sur l'icône ?



Nous voulons positionner le test en dessous :





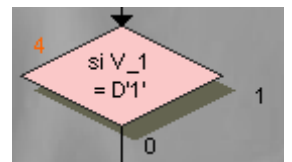
Choisir la variable V_1

Choisir le test =

Choisir la valeur D'1'

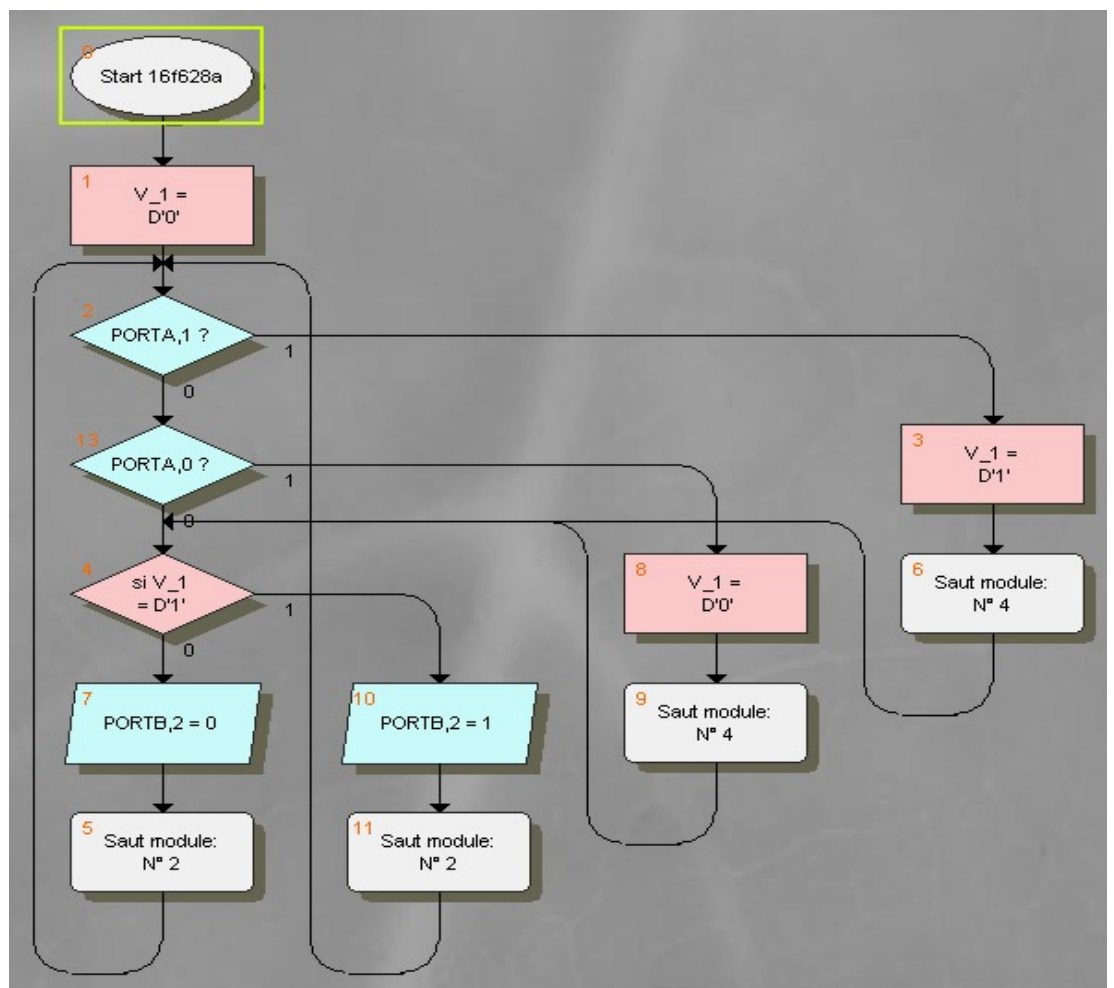
Valider par OK

Vous obtenez le module suivant :



Ce module se comprend simplement : le test se dirige vers 1 si la variable V_1 est égal à 1, le test se dirige vers 0 si la variable V_1 est différent de 1.

Compléter l'organigramme en prenant cet exemple :



Fonctionnement: à la mise en route, la variable V_1 se positionne à 0, sans appui sur les boutons poussoir, le programme tourne continuellement dans la boucle positionnant PORTB,2 = 0

Dès l'appui de la touche A1, la variable V_1 se positionne à 1, la boucle passe maintenant dans PORTB,2 = 1 jusqu'au moment où l'appui sur la touche A0 repositionne la variable V_1 à 0

