

I. Objectif

Découvrir les éléments simples d'une structure programmée.
Simuler un programme de MicroContrôleur.

II. Présentation du Microcontrôleur

1. Qu'est-ce qu'un Microcontrôleur ?

C'est un tout petit **composant** qui agit comme un ordinateur.

- Il **CALCULE** : toutes les opérations simples (+ - x /)
- Il **DECIDE** : toute la logique simple (et, ou, comparaison, test de valeur)
- Il a une mémoire pour garder : Un **PROGRAMME**
Le **RÉSULTAT** de ses **CALCULS**
- Il peut **DÉTECTER** grâce à ses **ENTRÉES** : des **BOUTONS**, des **CAPTEURS** etc...
- Il peut **COMMANDER** grâce à ses **SORTIES** : des **LEDS**, un **AFFICHEUR**, un **MOTEUR** etc...

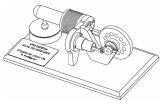


Il en existe de très simples avec peu de fonctions : 8 pattes, quelques pesos.
D'autres sont beaucoup plus complexes et puissants : centaines de pattes, millier de pesos.
Tout dépend de l'utilisation souhaitée

2. Où les trouvent-on ?

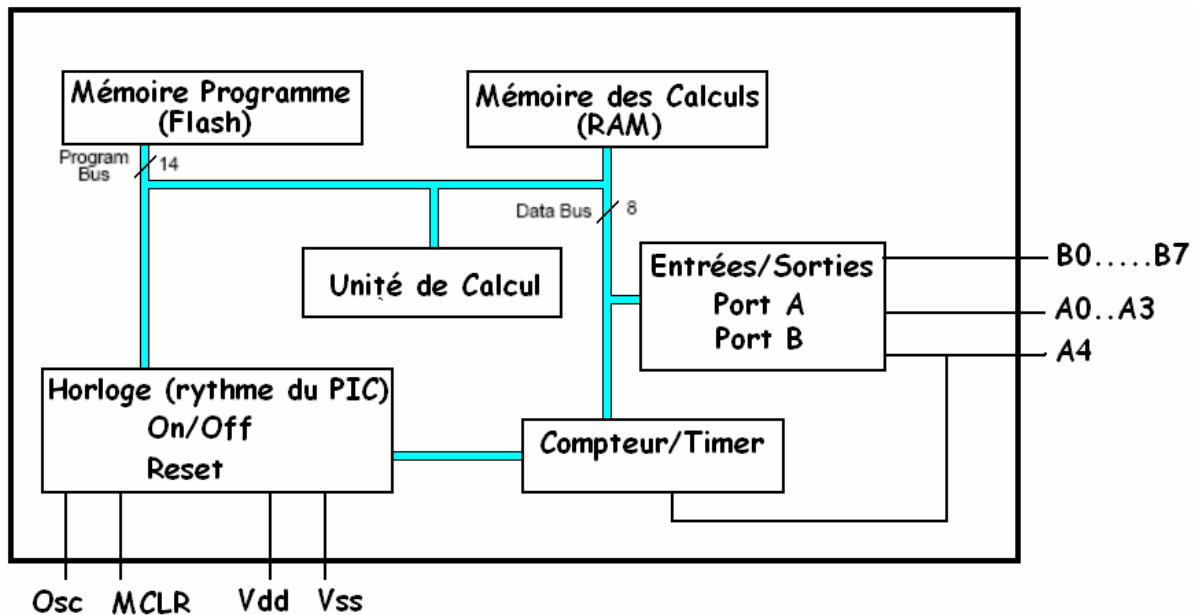
PARTOUT !

Besoin de différents modes de programmation ? Machine à laver, Four Micro-onde...
Besoin d'agir étape par étape ? Machine à café, Lave-Auto automatique...
Besoin de tester ? Détecteur de faux billets, Lecteur de carte, Tourniquets de metro, Alarmes...
Besoin de mesurer ? Thermomètres électroniques, Sondes météo...
Besoin d'afficher ? Appareils de mesures, Imprimante avec écran...
Besoin de mémoriser ? Télévision (réglage des chaînes)...
Besoin de traiter ? Décodeur mp3, Equaliseur d'autoradio...



3. Comment fonctionnent-ils ? ?

Structure Simplifiée du Microcontrôleur PIC 16F84a



Une Unité de Calcul est connectée à différents modules et gère l'ensemble :

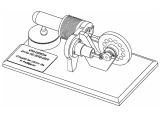
L'Unité de Calcul reçoit un ordre venant de la Mémoire de Programme et l'exécute.

Tout est écrit en binaire (0 ou 1 ; 0 volt ou 5 volt) : Les ordres codés et les nombres. Pour aller plus vite, des fils sont placés en parallèles et envoient les chiffres binaires en même temps. Ces « autoroutes » sont appelés BUS. Ici, le but des données est de 8 bits (8 fils)

Suivant l'ordre donné, l'Unité de Calcul :

- lit les valeurs des Ports d'Entrées, de la Mémoire des Calculs ou du Compteur/Timer.
- écrit son résultat dans la Mémoire des Calculs, les Ports de Sorties ou le Compteur/Timer.

Le tout est surveillé par une unité qui impose le Rythme (Horloge), la mise en marche et le Reset.

	<i>MÉCANIQUE - I.S.P.</i>	2AD
	Activité FlowCode	TP

III. Organigramme & Aide Informatique

1. Pourquoi un Logiciel de Programmation ?

Voici un extrait de Programme situé dans un Microcontrôleur :

La mémoire se compose de millions (milliards) de petits Switchs qui sont ouverts ou fermés. Cela se traduit par des 0 ou des 1 que le microprocesseur décode :

Ici, les 6 premiers sont le code d'un ordre (additionner, lire, écrire etc..)

Avec 6 chiffres binaires, je peux avoir $2^6 = 64$ ordres différents.

Ensuite les ordres ont besoins d'un nombre (dépend de l'ordre). Ce sont les 8 chiffres suivants.

Avec 8 chiffres binaires, je peux avoir un nombre de 0 à 255.

Si je veux un chiffre plus grand, je mets un ordre spécial qui lit les 16 chiffres suivants (de 0 à 65535)

Langage Machine
<pre> ... 011010 11100101 101001 01001110 010110 11011100 000111 00110011 101010 00101110 ... </pre>

IMAGINEZ ECRIRE UN PROGRAMME de 4000 LIGNES (4ko) !!!

Des programmes appelés « Compilateurs » nous aident à écrire ce code binaire avec des mots simples.

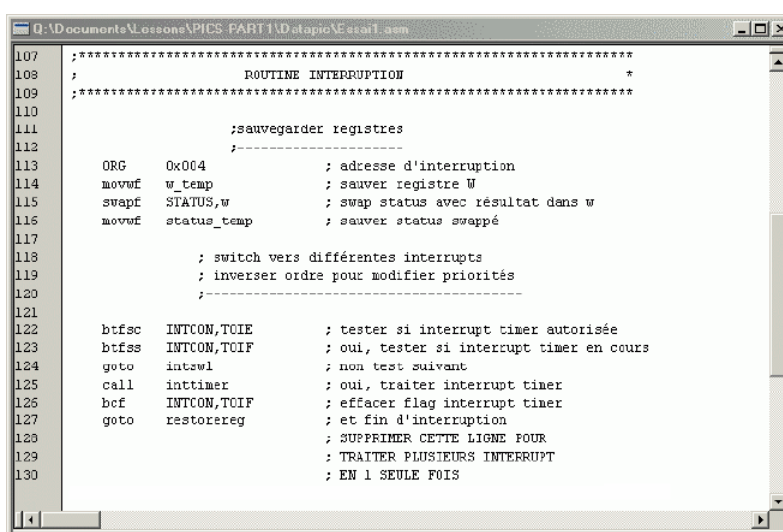
Voici un extrait de programme avec MPLab (logiciel Gratuit de Microchip)

Remarquez :

Les annotations pour ne pas se perdre.

Ce code appelé « Assembleur » est plus clair, mais encore des mots codés qu'il faut connaître.

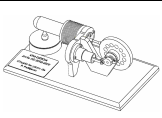
Avec l'« Assembleur », il faut aussi connaître la structure exacte du Microcontrôleur (mémoires spéciales etc...)



```

107 ;*****
108 ;          ROUTINE INTERRUPTION          *
109 ;*****
110
111           ;sauvegarder registres
112           ;-----
113   ORG     0x004           ; adresse d'interruption
114   movwf  w_temp          ; sauver registre W
115   swapf  STATUS,w        ; swap status avec résultat dans w
116   movwf  status_temp     ; sauver status swapé
117
118           ; switch vers différentes interrupts
119           ; inverser ordre pour modifier priorités
120           ;-----
121
122   btfsc  INTCON,TOIE     ; tester si interrupt timer autorisée
123   btfss  INTCON,TOIF     ; oui, tester si interrupt timer en cours
124   goto  intsw1          ; non test suivant
125   call  inttimer         ; oui, traiter interrupt timer
126   bcf   INTCON,TOIF     ; effacer flag interrupt timer
127   goto  restorereg      ; et fin d'interruption
128           ; SUPPRIMER CETTE LIGNE POUR
129           ; TRAITER PLUSIEURS INTERRUPT
130           ; EN 1 SEULE FOIS

```



2. Organigramme

Un Organigramme est un Schéma qui décrit une succession d'actions. Il se lit de Haut en bas :

Une Action peut être : Un calcul, une lecture d'entrée, envoyer un signal sur une sortie, un test ($a = 5$?) etc...

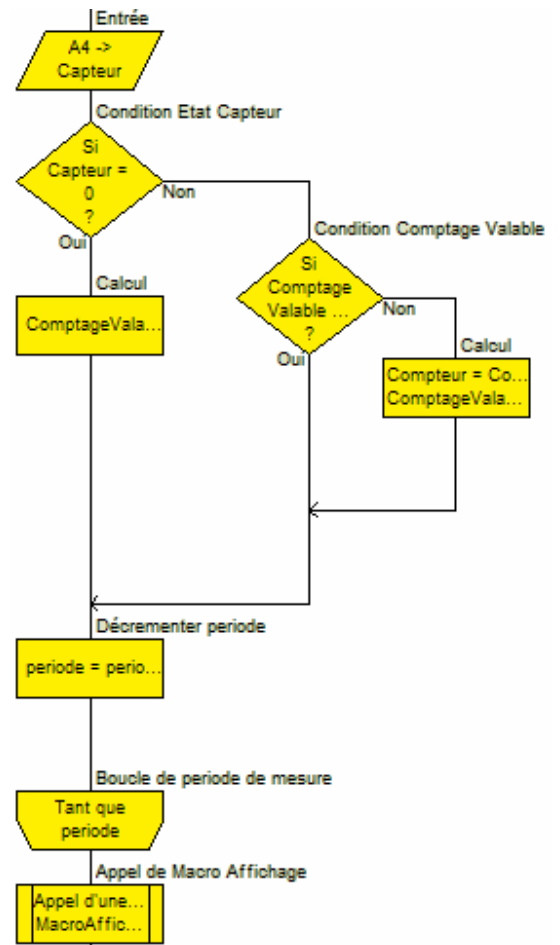
Les actions se font les unes après les autres.

Un test (oui/non) divise le chemin en deux.

On peut ainsi **décrire toutes les tâches** que doit effectuer un microcontrôleur, dans quel ordre il doit les faire et sous quelle **condition**.

Exemple :

La machine à café lance la procédure du café choisi à condition qu'elle détecte les 6 pesos nécessaires.



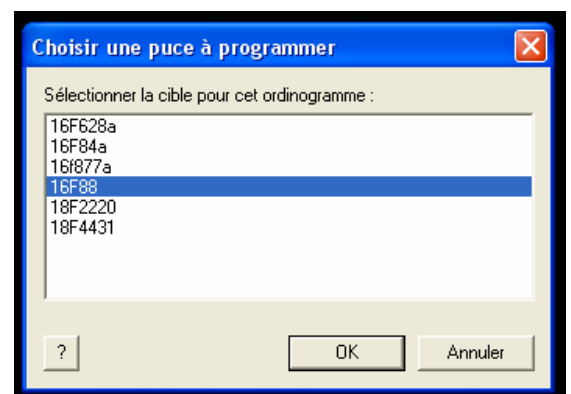
IV. FlowCode

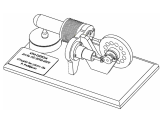
1. Présentation de Flowcode

Pour commencer, FlowCode demande **quel Microcontrôleur** va être utilisé :

On choisit le 16F88

Il est identique au 16F84a
Mais avec plus de mémoire





ENVIRONNEMENT FLOWCODE

Créer le programme et l'envoyer vers le PIC

Simuler le programme

Organigramme de Travail

Eléments d'Organigramme (à faire glisser)

Composants que peut gérer le Microcontrôleur (Exemple: Capteur en Entrée et Afficheur en Sortie)

Puce

16F84A			
RA2	1	18	RA1
RA3	2	17	RA0
RA4/T0CKI	3	16	OSC1/CLKIN
MCLR	4	15	OSC2/CLKOUT
Ves	5	14	VDD
RB0/INT	6	13	RB7
RB1	7	12	RB6
RB2	8	11	RB5
RB3	9	10	RB4

Microcontrôleur Choisi (PIC)

Appuyez sur F1 pour obtenir de l'aide

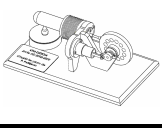
Zoom actuel = 75%

NUM

FlowCode vous aide à faire l'Organigramme (icônes jaunes).

Il vous place une case « début » et une case « fin ».

Vous pouvez insérer d'autres cases entre les deux, en faisant glisser les icônes jaunes.

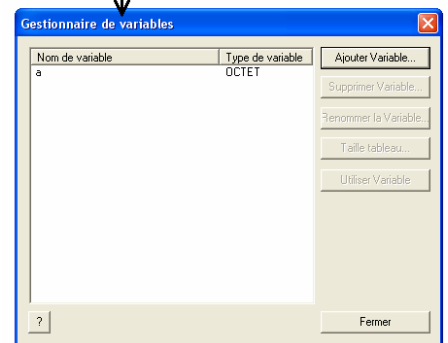
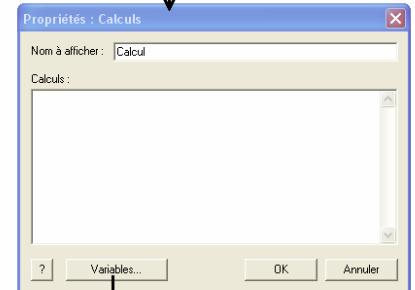


Activité FlowCode

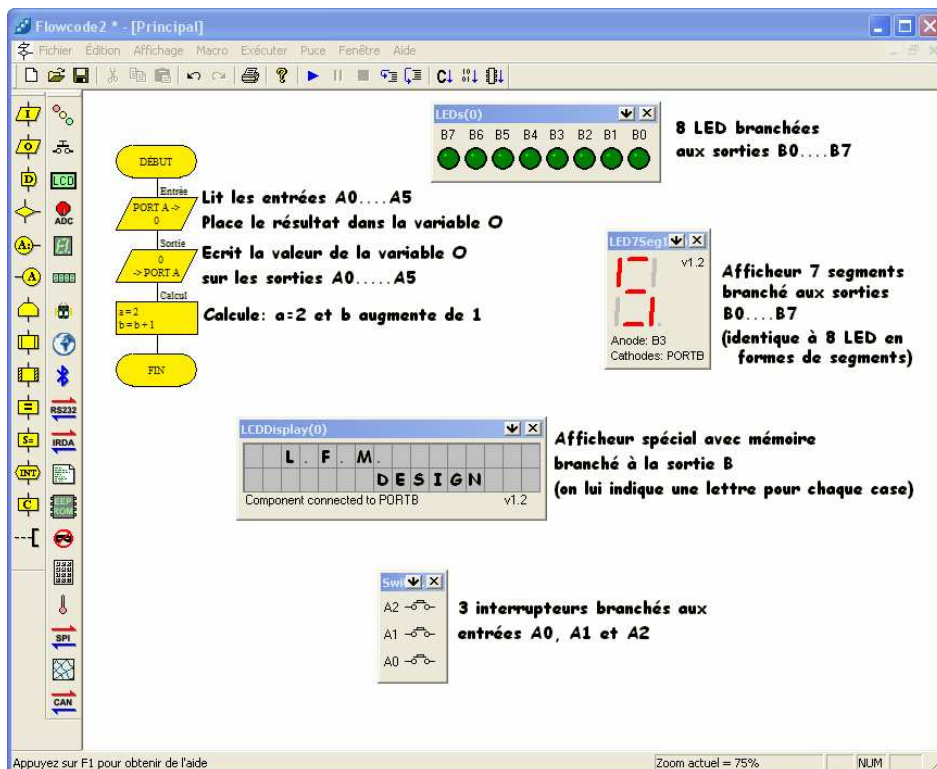
Pour chaque case, vous pouvez cliquer «Propriétés» : Cela permet d'écrire le test, le calcul, la valeur de sortie etc...

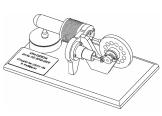
NOTE :

Pour certaines propriétés, vous avez l'option «variable». Elle permet de déclarer toutes les variables qui seront dans vos calculs et tests.



Vous pouvez faire apparaître des LED, un Afficheur, des Boutons etc...

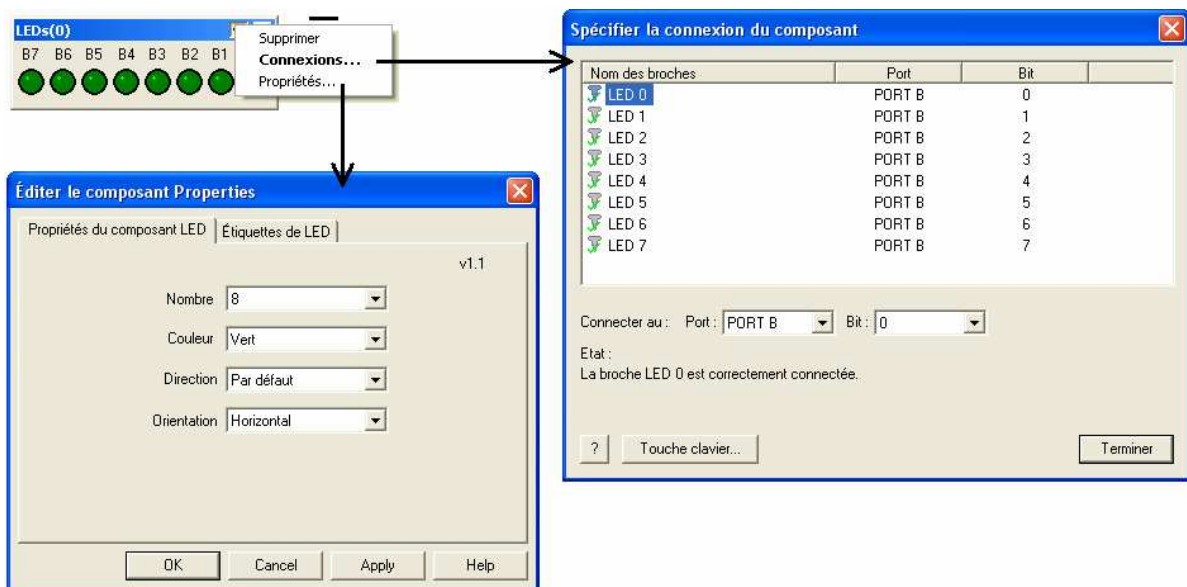




Pour chacun de ces composants périphériques, une petite flèche vers le bas permet de choisir :

- les **propriétés** : Nombre de diodes, couleur des diodes, donner un nom à chaque diode...
- la **connexion** avec le PIC : On voit ici que la diode 0 (LED 0) est reliée à la patte n°0 du port B. La diode 7 est sur la patte n°7 du port B.

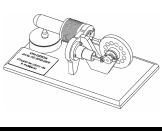
Notre PIC 16F88 possède 2 ports (A et B). On les choisit comme entrée ou sortie. Le port A possède 5 pattes et le port B 8 pattes



Pour FlowCode, quel est l'intérêt de placer des composants périphérique autour du PIC ?

Cela permet à FlowCode de **SIMULER** votre programme et voir comment le microcontrôleur **réagit** avec les composants périphériques. Si vous appuyez un bouton, quelles LED s'allume etc...

Lorsque les erreurs sont **corrigées**, le programme sera traduit (compilé), puis envoyé vers un **vrai** microcontrôleur PIC qui aura de **vrais** composants autour de lui.



V. Activités FlowCode

1. Diode Clignotante

Cet exercice doit vous permettre de **découvrir** les Fonctions Principales de FlowCode.

Appuyez sur F1 pour obtenir de l'aide

Zoom actuel = 75%

NUM

- **CHOISIR** le Composant 16F84a (demandé au début ou par le menu « puce »)
- **POSER** les cases de l'Organigramme ci-dessus. Les A sont des jonctions : quand on arrive sur un A, on saute à l'autre A. Ne touchez pas aux propriétés des cases.
- **PLACER** un ensemble de 8 diodes.
- **CONNECTER** la 1^{ère} diode sur une des pattes (au choix) du port B.
- **PLACER** dans la 1^{ère} cellule Sortie : valeur 1 sur le port B. Sélectionner « bit unique » qui veut dire que le 1 sera placé sur une seule patte. Choisir la patte (celle où sera la diode)
- **PLACER** dans la 2^{ème} cellule Sortie : valeur 0 sur le port B. Sélectionner aussi « bit unique ». Choisir la même patte.
- **SIMULER** avec le bouton lecture (triangle bleu)

2. Diode Clignotante avec une Variable

Cette fois, on crée une variable qui s'appelle « a ».
On pose a = 1 au départ.

- **FAIRE** ce nouvel organigramme en gardant la LED de l'exercice précédent.
- **SIMULER.**
- **CHANGER** la connexion de votre diode pour la placer sur la patte A2. Que faut-il aussi changer dans l'organigramme pour que la diode clignote correctement.

.....
.....


- **CHANGER** la 2^{ème} pause de votre organigramme avec 0,2secondes (case jaune juste avant la jonction A). Que se passe-t-il ?

.....
.....

On peut raccourcir le programme de la manière suivante :

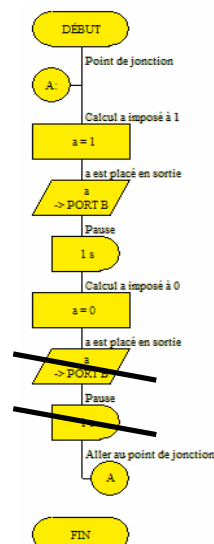
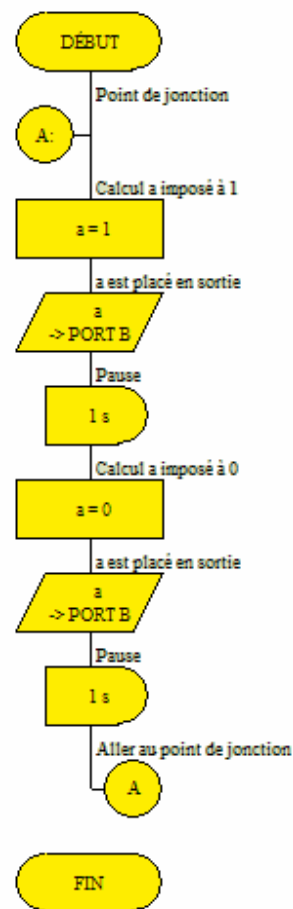
A la place du calcul a = 0 , on écrit : $a = 1 \text{ XOR } a$ (Attention aux majuscules)
Cette commande fait passer a de 0 vers 1 ou de 1 vers 0.
Je peux donc supprimer les 2 cases rayées (ci-contre)

- **SIMULER.** Ca clignote ?

Il y a sûrement une erreur, il faut déplacer une case. Utiliser la simulation « Pas à Pas »  pour exécuter les instructions une par une.

- **QUEL DEPLACEMENT** de case faut-il ? **POURQUOI ?**

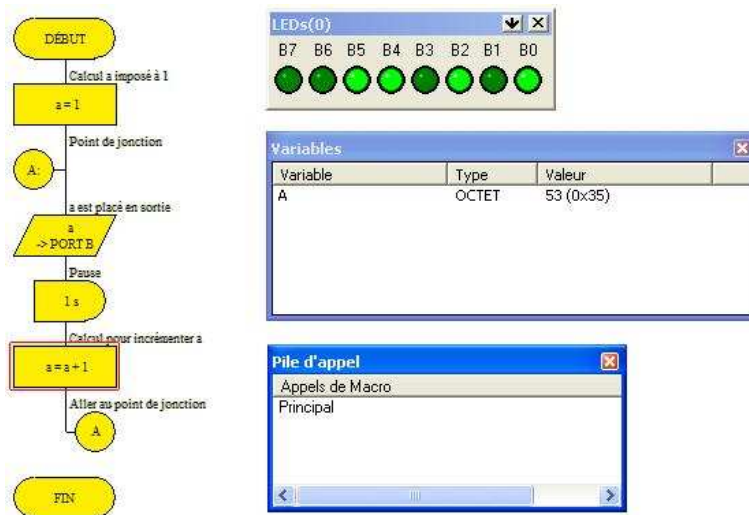
.....
.....



3. Comptage Binaire

Cet exercice va vous permettre de comprendre ce qui se passe si on écrit un nombre dans le port complet.

- **POSER** 8 diodes toutes connectées.
- **REFAIRE** cet organigramme.
- **SIMULER** en cliquant « Lecture ».
- **CLIQUER** l'icône « Pause » à côté.



En cliquant sur « Pause », apparaissent deux tableaux :

- L'un indique le programme en cours.
 - L'autre indique les valeurs des variables (ici, seulement « a »)
- **RELANCER** la simulation et cliquer sur pause à chaque changement des diodes.
 - **QUELLE VALEUR** de « a » allume seulement la diode B1 ?.....
 - **QUELLE VALEUR** de « a » allume seulement la diode B2 ?.....
 - **QUELLE VALEUR** de « a » allume seulement la diode B3 ?.....
 - **QUELLE VALEUR** de « a » allume seulement les diodes B0, B2 et B3 ?.....

Une diode allumée signifie qu'elle reçoit un 1, une diode est éteinte si elle reçoit un 0.

Si je regarde les diodes **B7 à B0** sur cette page, je lis : **0-0-1-1-0-1-0-1**.

Ce sont les valeurs imposées par le microcontrôleur car a = 53

- **RELANCER** la simulation et cliquer sur pause à chaque changement des diodes. A chaque pause, écrivez dans le tableau les valeurs imposées par le microcontrôleur suivant « a ».

Valeur de « a »	Suite de 0 et de 1 en regardant de B7 à B0	Valeur de « a »	Suite de 0 et de 1 en regardant de B7 à B0
1		9	
2		10	
3		11	
4		12	
5		13	
6		14	
7		15	
8		16	