

1. Présentation du Microcontrôleur

1.1. Objectif de la séance :

Cette séance permettra de comprendre le fonctionnement d'un microcontrôleur (ports, entrées, sorties), de concevoir et de tester son programme de manière graphique ainsi que d'implémenter le programme dans la mémoire et de réaliser les tests de fonctionnement.

1.2. Mise en situation :

1.2.1. Le microcontrôleur :

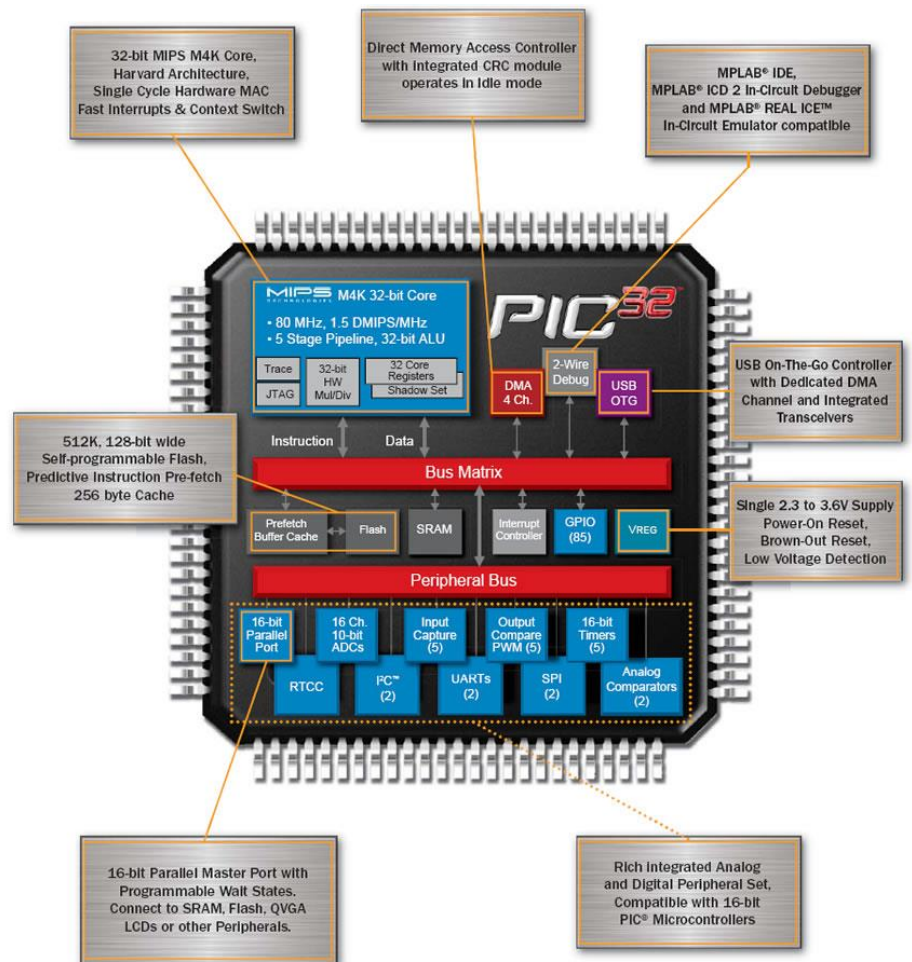
Un **microcontrôleur** est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur :

- **processeur** : il interprète et exécute les instructions
- **mémoires** :
 - ROM : mémoire morte pour le programme,
 - RAM : mémoire vive pour les données),
- unités périphériques,
- interfaces d'entrées-sorties.

Les microcontrôleurs se caractérisent par rapport aux **microprocesseurs polyvalents** utilisés dans les ordinateurs personnels par :

- un plus haut degré d'intégration,
- une plus faible consommation électrique,
- une vitesse de fonctionnement plus faible (quelques MHz à quelques centaines de MHz),
- un coût réduit.

Le microcontrôleur est donc un circuit programmable, capable de gérer des périphériques, des entrées et des sorties logiques, numériques et analogiques.



Les **microcontrôleurs PIC** (ou **PICmicro** dans la terminologie du fabricant) forment une famille de microcontrôleurs de la société Microchip.

Le nom PIC n'est pas officiellement un acronyme, bien que la traduction en « Peripheral Interface Controller » (contrôleur d'interface périphérique) soit généralement admise.

1.3. Le microcontrôleur PIC 16F84A :

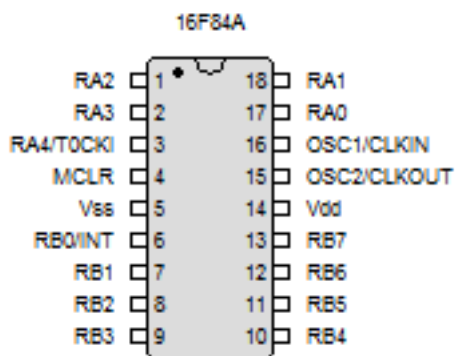


Le 16F84A, possède 18 broches.

Les broches 14 et 5 sont les broches d'alimentation du μC , respectivement **Vcc** (+5V) et **Vss** (0V).

La broche 4, MCLR correspond au reset global du circuit. Il est actif au niveau bas.

Les broches 15 et 16 correspondent à l'horloge de fonctionnement du circuit. Cette horloge est réalisée soit avec un oscillateur (quartz) soit à l'aide d'une horloge externe. **La valeur de la fréquence de l'horloge devra être indiquée lors de la programmation du circuit.**



Le **PORTB**, comprend un octet (BYTE) de donnée (DATA) de RB0 à RB7. Chaque bit de cet octet peut être utilisé et configuré comme une entrée (I) ou comme une sortie (O), il est bidirectionnel. Le bit RB0 peut de plus, être utilisé comme un signal d'interruption (INT) externe.

Le **PORTA** est un octet tronqué, car il ne comprend que les 5 premiers bits de l'octet de RA0 à RA4. Chaque bit de ce PORT peut également être configuré en I/O. Le bit RA4 peut de plus être utilisé en qualité de timer0 (temporisateur/compteur)

1.4. Le logiciel FLOWCODE :

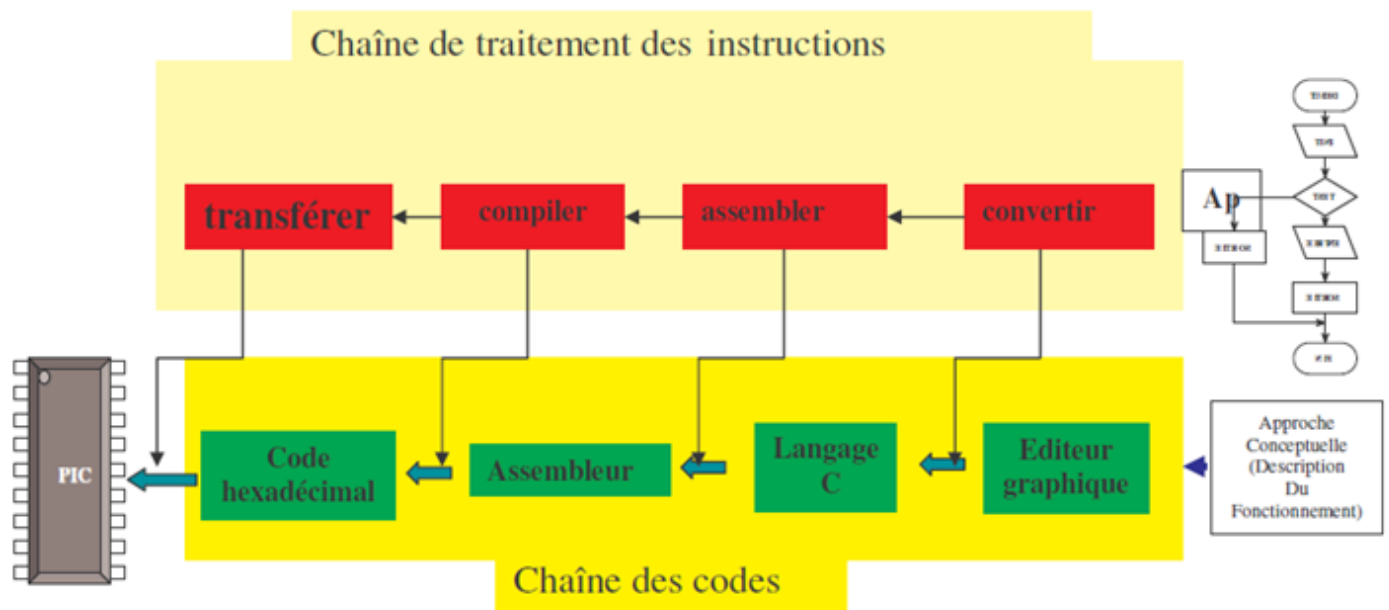


Flowcode est un logiciel graphique de programmation spécialisé pour les $\mu\text{contrôleurs}$ de la famille PIC du fabricant *Microchip*.

Les graphiques utilisés par ce logiciel sont des algorithmes ou ordinogrammes, ils permettent la simulation du programme et la compilation du programme en code Hexadécimal (.Hex) compréhensible par le PIC.

Note : Il est également possible d'obtenir un code en langage C compréhensible par l'homme.

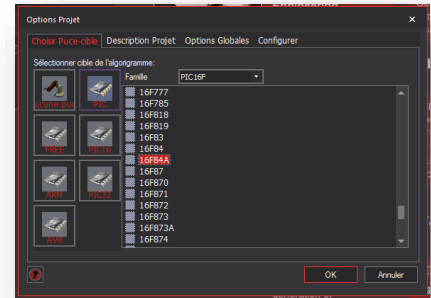
Approche globale de la programmation



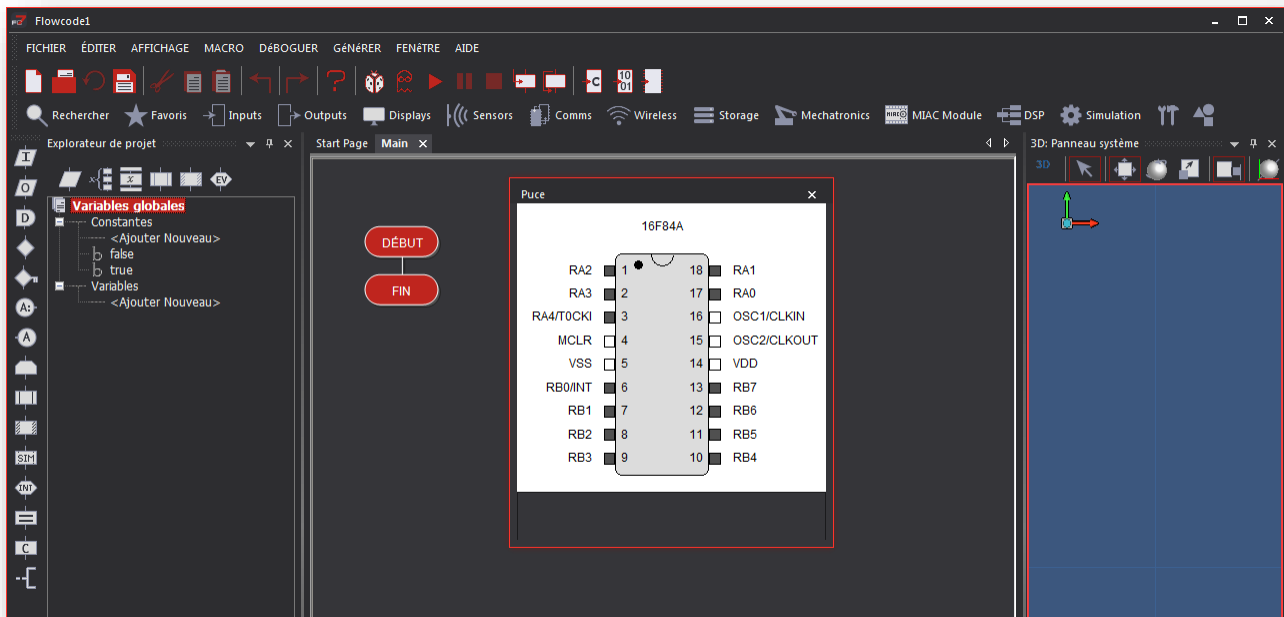
2. Programme n°1: allumer une diode grâce à un interrupteur

2.1. Affectation des entrées/sorties :

- ☞ Lancer le logiciel **FLOWCODE V7**.
- ☞ Cliquer sur "Fichier", "Nouveau projet"
- ☞ Choisir le µcontrôleur 16F84A



Vous obtenez la page suivante à l'écran :



Dans ce premier exercice nous allons réaliser un programme qui permet d'allumer une diode en agissant sur un interrupteur bistable.

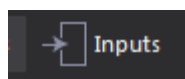
L'interrupteur S0 sera raccordé sur le bit 0 du port B (RB0)



La LED1 sera raccordée sur le bit 1 du port A (RA1)



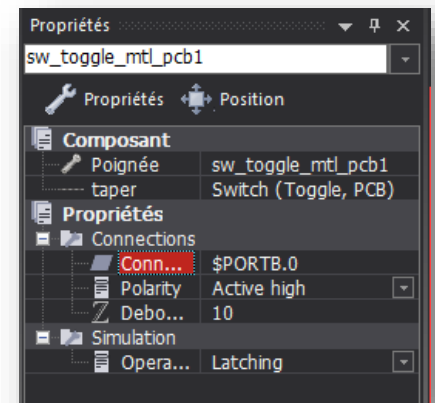
Insérer un interrupteur à l'aide du menu déroulant "Inputs".



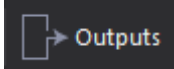
Cliquer sur "Add to 2D dashboard panel"

Affecter l'interrupteur S0 au bit 0 du port B (RB0).

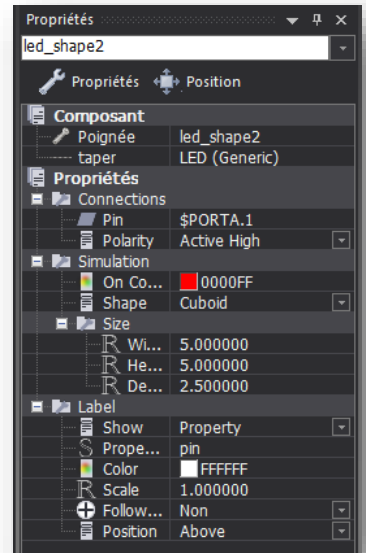
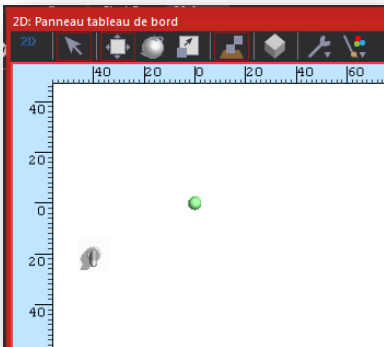
Clic droit sur le "switch", choisir "propriétés". cliquer sur "connections"



Insérer la diode à l'aide du menu déroulant "Output"



Affecter, en suivant la même procédure, une DEL rouge au bit 1 du PORTA (RA1).



2.2. Recherche de l'algorithme :

Pour écrire le programme, il faut traduire sous forme d'algorithme la phrase suivante :

Si l'interrupteur S0 est actionné (poussé vers le bas) alors on allume la DEL1, sinon, on éteint la DEL1.

Q1: Compléter le tableau des variables en barrant la réponse fautive:

S0=0	Interrupteur en position haute / basse
S0=1	Interrupteur en position haute / basse
DEL1=0	Diode allumée / éteinte
DEL1=1	Diode allumée / éteinte

Q2: Compléter l'algorithme ci-dessous:

Si S0= _____
 alors _____ DEL1 = _____
 sinon DEL1 = _____

Q3: Compléter l'algorithme ci-dessous en prenant en compte les adresses RBO et RA1 des entrées et sorties

S0 = _____ // On charge dans le variable S0 la valeur du bit 0 du port B
 Si S0 = _____ alors // On effectue un test sur le contenu de la variable S0
 DEL1 = _____ // On écrit la valeur souhaitée dans la variable DEL1 en fonction du test
 Sinon
 DEL1 = _____ // On écrit la valeur souhaitée dans la variable DEL1 en fonction du test
 RA1 = _____ // Le bit 1 du port A prend la valeur de la variable DEL1.

2.3. Traduction de l'algorithme en algorithme :

Pour réaliser l'algorithme, il faut utiliser :



une icône d'entrée



une icône de décision



une icône de sortie



des icônes de calcul

Définir les variables S0 et DEL1

Créer Nouvelle Variable

Nom Nouvelle Variable: S0

Valeur Initiale: 0

Description :

Type Variable:

- Bool (soit vraie, 1 soit fausse, 0)
- Octet (nombre allant de 0 à 255)
- Entier (nombre entre -32768 et 32767)
- UInt (nombre compris entre 0 et 65535)
- Long (nombre compris entre -2147483648 et 2147483647)
- ULong (nombre compris entre 0 et 4294967295)
- Chaîne de caractères (longueur par défaut = 20)
- Virgule flottante
- Identificateur d'objet

OK Annuler

Éditer Nom Variable

Saisir un nouveau nom de variable: DEL1

Valeur Initiale: 0

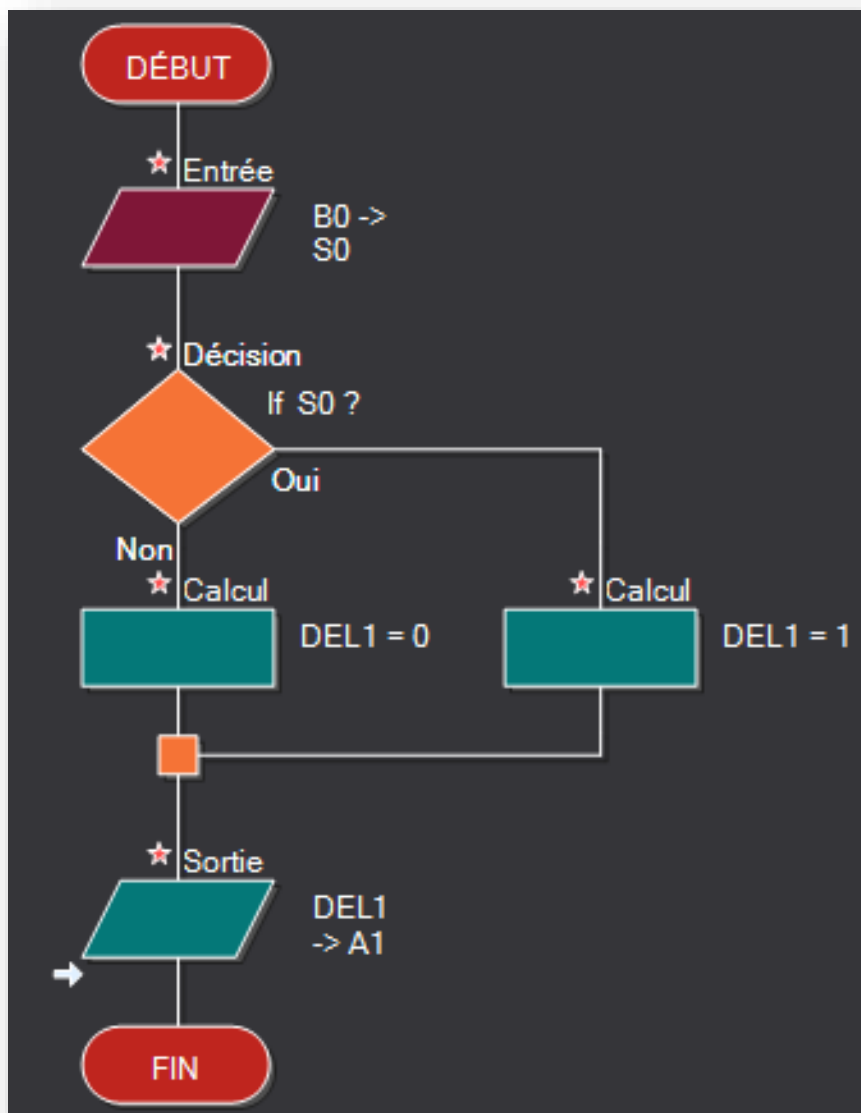
Description :

Type Variable:

- Bool (soit vraie, 1 soit fausse, 0)
- Octet (nombre allant de 0 à 255)
- Entier (nombre entre -32768 et 32767)
- UInt (nombre compris entre 0 et 65535)
- Long (nombre compris entre -2147483648 et 2147483647)
- ULong (nombre compris entre 0 et 4294967295)
- Chaîne de caractères (longueur par défaut = 20)
- Virgule flottante
- Identificateur d'objet

OK Annuler

Etablir l'algorithmme ci-dessous (accès aux différents pictogrammes par clic droit) et insérer les commentaires nécessaires à sa compréhension :

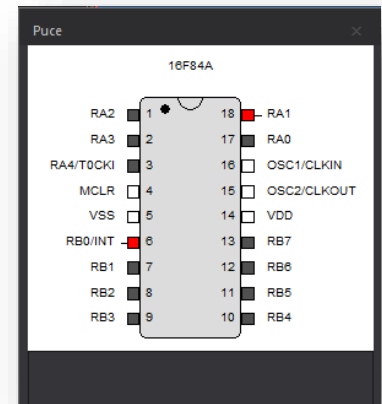


2.4. Simulation :

Exécuter la simulation en exécutant pas à pas les instructions du programme grâce au bouton :



Vous pouvez voir l'état des ports dans la fenêtre "Puce" :
Exemple : ici RA1 et RB0 sont à 1 (en rouge).



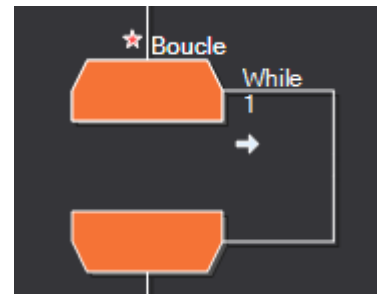
Q4. Que constatez-vous ? Le fonctionnement correspond t-il à celui attendu ?

Comme vous l'avez remarqué, le programme ne fonctionne pas !!!!!

Et oui, le programme ne s'exécute qu'une fois, si l'interrupteur était en position haute au début du programme, la DEL1 est restée éteinte, et s'il était en position basse, la diode s'est allumée et on ne peut plus l'éteindre.



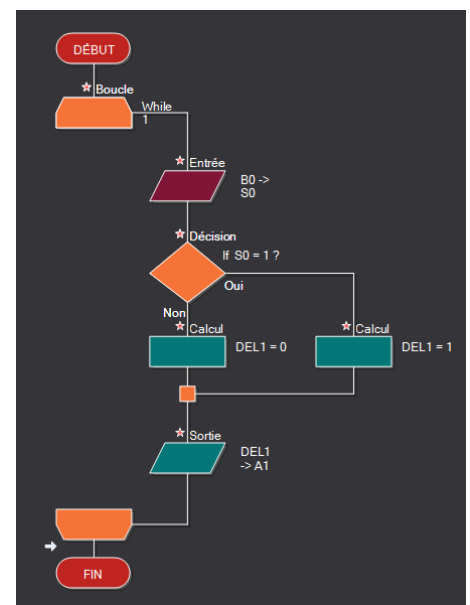
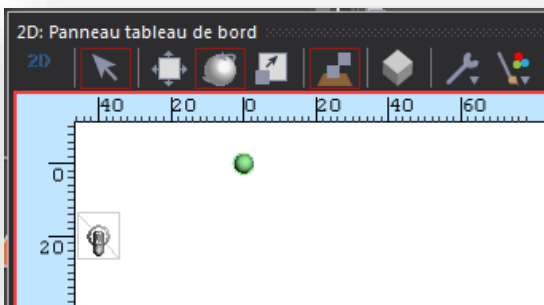
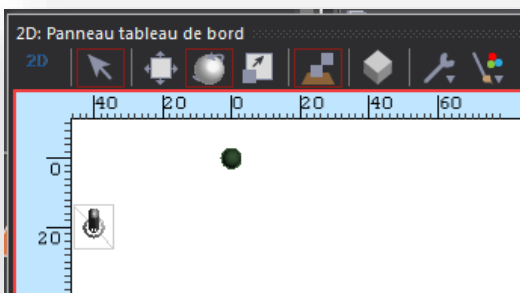
Il faut ajouter une boucle infinie à ce programme, avec l'icône ci-contre :



Réaliser la modification, exécuter le programme en simulation :



En changeant l'état de l'interrupteur sur le panneau 2D, vous verrez l'état de la diode changer.



3. Réalisation d'un clignoteur :

En reprenant la structure de l'algorithme précédent, concevoir le programme tel que :

Si l'interrupteur S0 est actionné, la DEL1 doit clignoter avec un allumage de 400ms, une extinction de 200ms, sinon, la DEL1 reste éteinte.



Note : Fonction temporisation: pictogramme « Delay ».

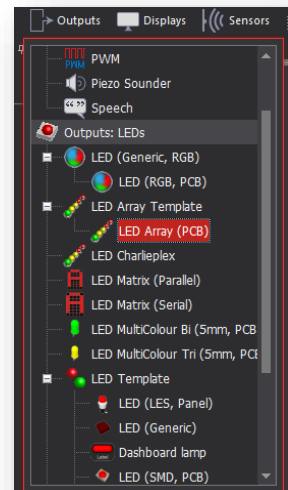
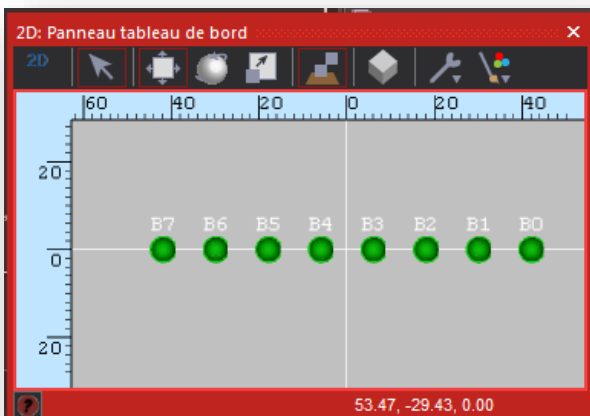
Q5. Modifier l'algorithme précédent et insérer les commentaires nécessaires à sa compréhension.

Q6. Réaliser la simulation et la faire valider.

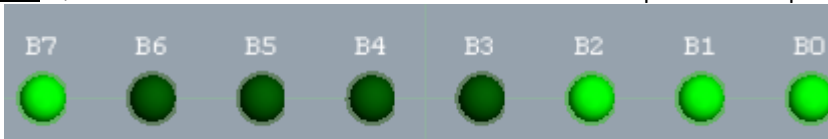
4. Programmer l'allumage d'une rampe de diodes

Ouvrir un nouveau projet, choisir le microcontrôleur 16F84A.

Cliquer sur « Output » et sélectionner « LED Array »



Q7. Quelle valeur faut-il mettre dans l'octet associé à la rampe de diodes pour afficher:



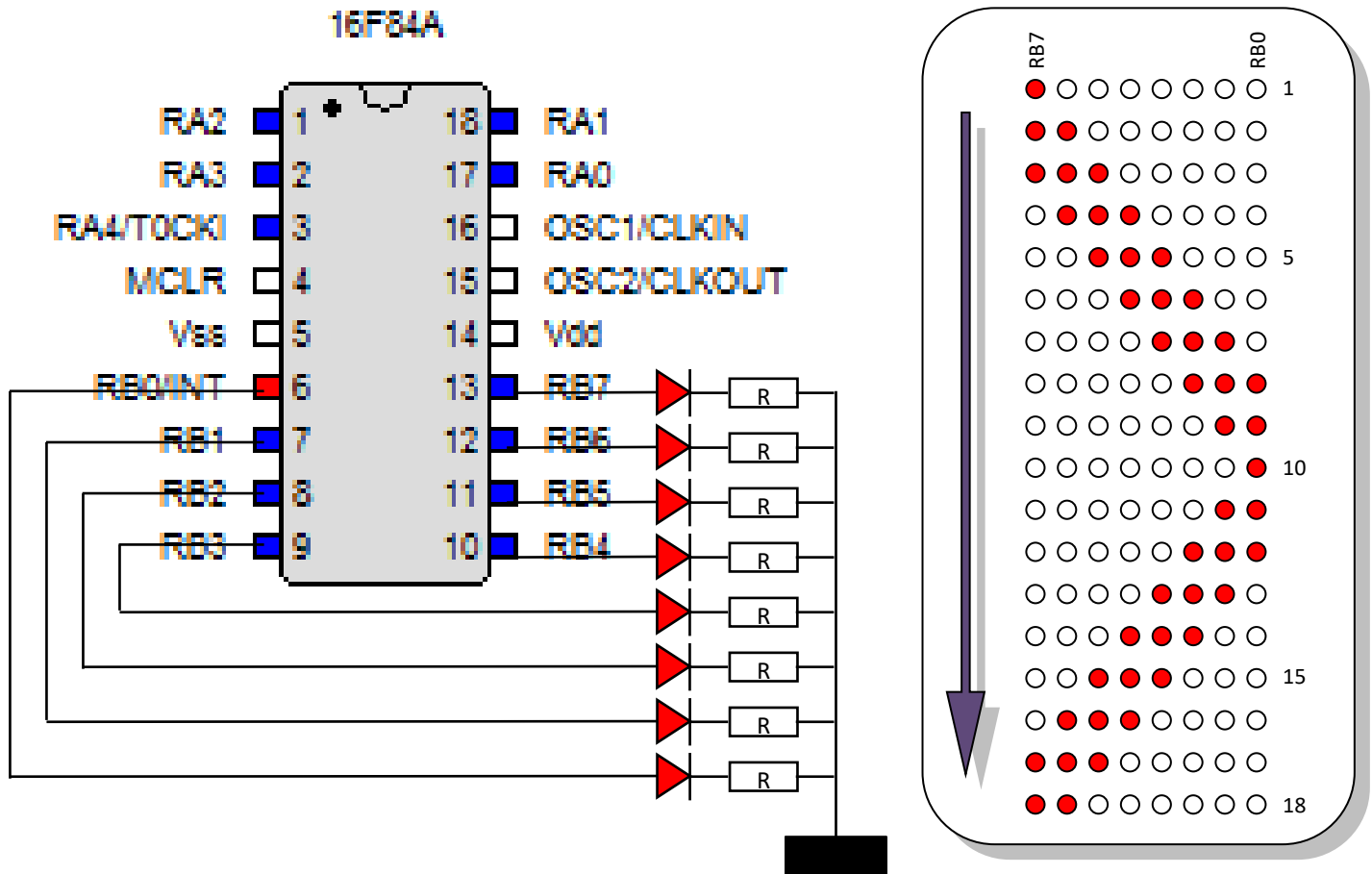
Q8. Quelles sont les diodes allumées lorsque l'on met l'octet à la valeur:57

B7	B6	B5	B4	B3	B2	B1	B0
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>


5. Chenillard :

Nous allons maintenant réaliser la **programmation** d'un **chenillard** :

Les LEDS sont connectées comme suit sur le Port B et on donne le **cycle qui devra être répété indéfiniment** :



5.1. Conseils pour la conception :

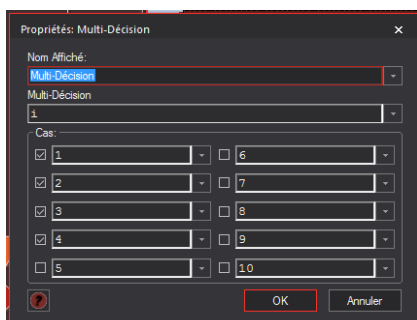
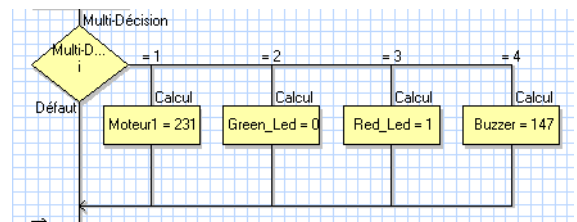
- Utiliser une structure **multi-décision**. 
- Travailler sur les **octets** et non plus uniquement sur les bits.
- Utiliser une **fonction** (macro) pour l'allumage des différentes LEDS.

Multi-décision :

Utiliser une variable qui s'incrémente (+1) ou se décrémente (-1) à l'aide de la fonction calcul. En fonction de la valeur de cette variable, allumer certaines LEDS (mot binaire à appliquer sur le port) grâce à la fonction multi-décision.

Dans l'exemple ci-contre, la variable s'appelle "i" et peut prendre les valeurs de 1 à 4.

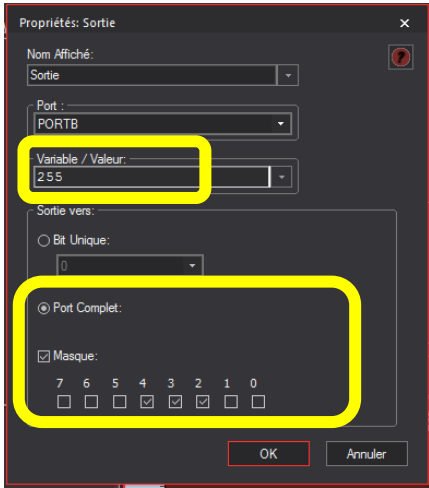
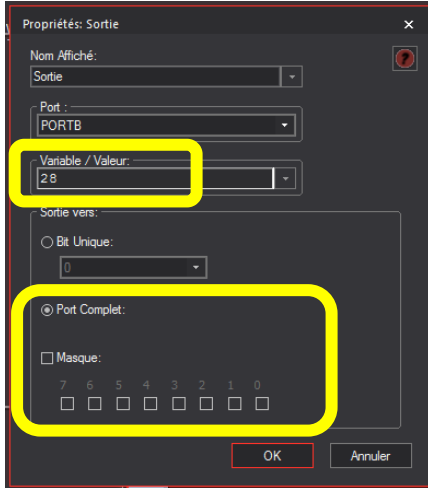
En fonction de sa valeur, une décision est prise.



Q9: De combien de cas différents avez-vous besoin pour réaliser le programme?

5.2. Travail sur les octets :

Exemple pour la ligne 6 ou la ligne 14 du cycle :

1 ^{ère} méthode :	2 ^{ème} méthode (à privilégier) :
On met tous les bits à 1 (valeur 255) et on utilise un masque.	On met à 1 uniquement les bits que l'on souhaite sans utiliser de masque, uniquement par calcul de la valeur de l'octet.
	

Rappel sur la numération

Bit	a7	a6	a5	a4	a3	a2	a1	a0
Poids	7	6	5	4	3	2	1	0
Valeur décimal	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Exemple : Lignes 6 et 14.	0	0	0	1	1	1	0	0
	4 + 8 + 16 = 28 Donc il faudra imposer une valeur de 28 au port B complet.							

5.3. Création d'une macro (fonction) :

Afin d'améliorer la lisibilité du code nous allons utiliser une macro qui se chargera d'allumer les LEDs en fonction de l'évolution du programme principal.



Q10: Réaliser l'algorithme, insérer les commentaires nécessaires à sa compréhension et faire la simulation.

Q11: Recopier votre algorithme et la macro associée.

