

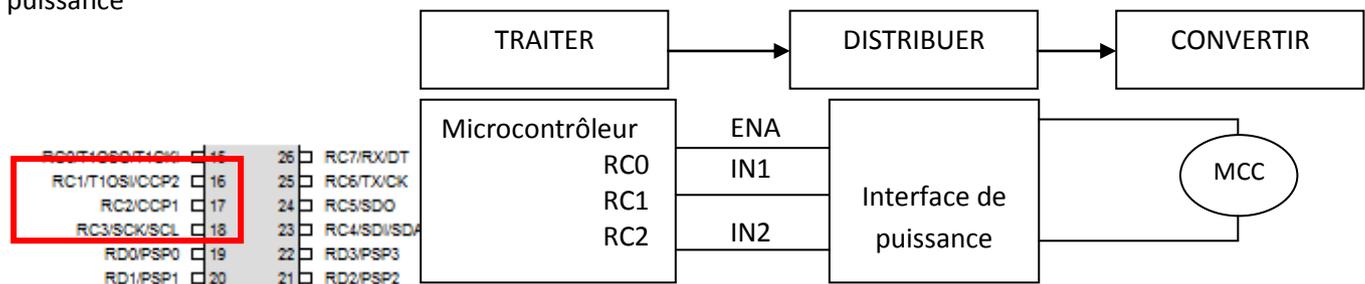


**Logiciel** : Flowcode

**Composant** : 16F877A

### Pilotage d'un moteur à courant continu

Le pilotage d'un moteur à courant continu à partir d'un microcontrôleur se fait par l'intermédiaire d'une interface de puissance



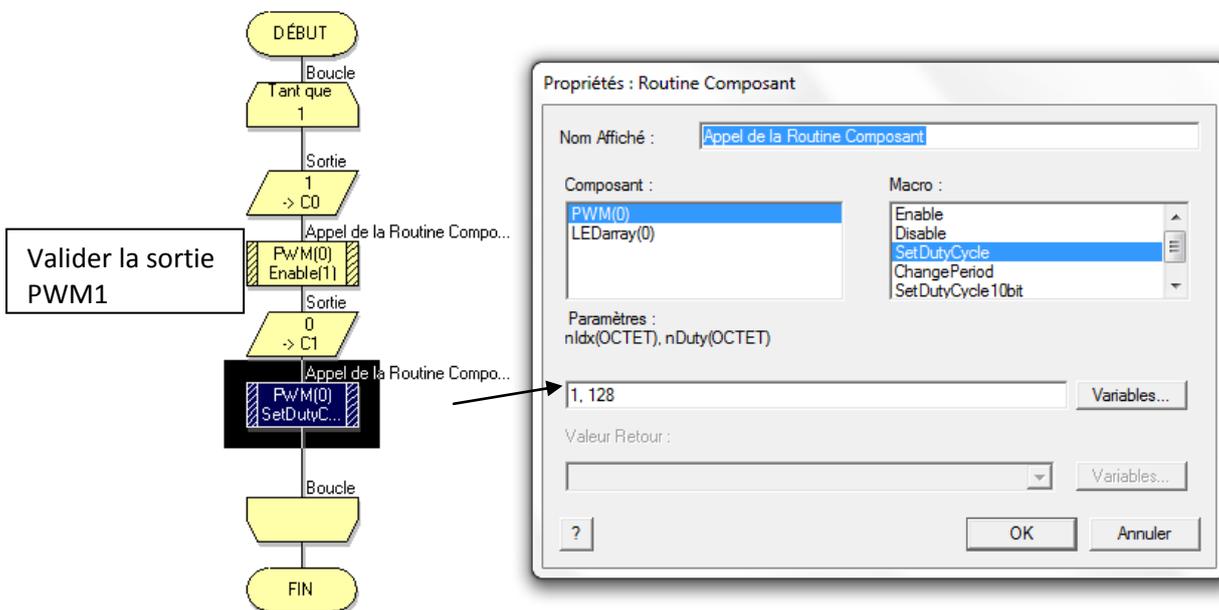
- Les sorties du microcontrôleur utilisées sont :
- RC0 : Validation de l'interface de puissance
  - RC1 et RC2 : Sortie PWM du microcontrôleur

L'interface de puissance est un pont en H piloté par trois entrées dont voici la table de vérité :

ENable	IN1	IN2	Etat moteur
0	X	X	Arrêt
1	1	0	Sens 1
1	0	1	Sens 2

Pour piloter un moteur à courant continu sous Flowcode, on utilise l'outil PWM (Pulse Width Modulation) dans l'onglet Mechatronics, qui permet de générer les signaux pilotant l'interface de puissance qui fournit l'énergie électrique au moteur à courant continu.

**Q1.** A l'aide de la notice de Flowcode, réaliser le programme suivant puis lancer la simulation :

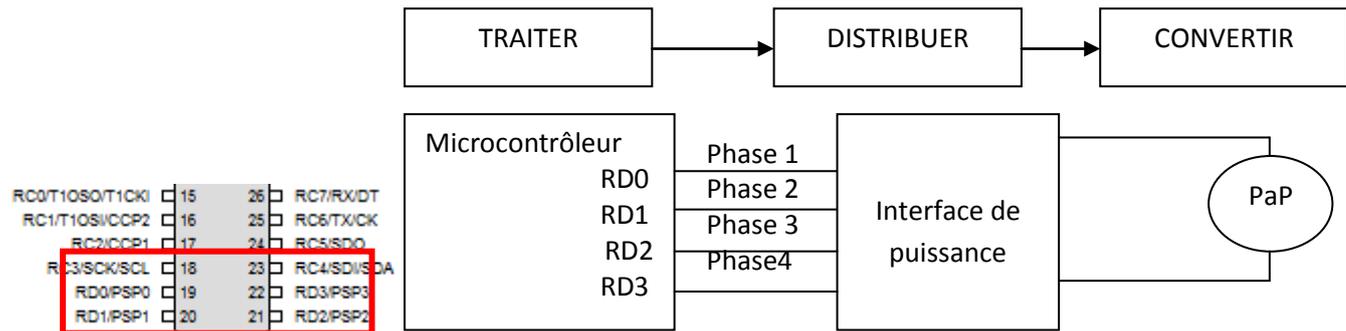


Q2. Télécharger  le programme dans le pic et le tester. Commenter les résultats de la simulation.

Q3. Réaliser un programme qui permet de faire tourner le moteur à 75% de sa vitesse dans le sens 1 si on n'appuie pas sur l'interrupteur A0 et à 50% de sa vitesse dans le sens 2 si on appuie sur A0. Vous utiliserez l'outil de décision .

### Pilotage d'un moteur pas à pas

Le pilotage d'un moteur à pas à pas à partir d'un microcontrôleur se fait également par l'intermédiaire d'une interface de puissance



Les sorties du microcontrôleur utilisées sont :

- RD0 : Pilotage de la phase 1 du moteur PaP
- RD1 : Pilotage de la phase 2 du moteur PaP
- RD2 : Pilotage de la phase 3 du moteur PaP
- RD3 : Pilotage de la phase 4 du moteur PaP

Table de vérité de l'interface de puissance :

	Phase 1	Phase 2	Phase 3	Phase 4
1	1	0	0	1
2	0	1	0	1
3	0	1	1	0
4	1	0	1	0

Pour piloter un moteur pas à pas sous Flowcode, on utilise l'outil STEPPER , qui permet de générer les signaux pilotant l'interface de puissance qui fournit l'énergie électrique au moteur pas à pas.

Q1. A l'aide de la notice de Flowcode, réaliser le programme suivant puis lancer la simulation  :

The screenshot shows a Flowcode program on the left and a 'Propriétés de : Éditer Composant' dialog box for a stepper motor on the right. The dialog box has several red circles highlighting specific settings:

- 'Full steps/rev' is set to 20.
- 'Drive mode' is set to 'Full Step (2-2)'. The 'Phase Patterns' table below it shows:
 

	A	B	C	D
0	1	0	0	1
1	0	1	0	1
2	0	1	1	0
3	1	0	1	0
- 'Connection' is set to 'Bipolar'.

Le moteur tourne de fait  $1.8^\circ$  par pas.

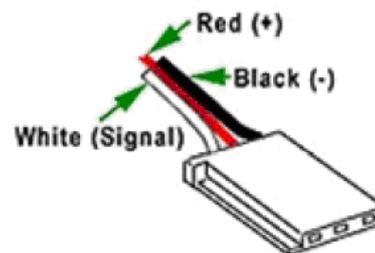
**Q2.** Quel est le nombre de pas par tour ? Renseigner « Fullstep » avec cette valeur. Remplacer la temporisation de 1s par 10ms puis télécharger  le programme dans le pic et le tester. Commenter les résultats de la simulation.

**Q3.** Changer la valeur de la temporisation afin de trouver la valeur (plage) de fréquence de rotation maximum.

**Q4.** Réaliser un programme qui permet de faire tourner le moteur à une fréquence de 100 Hz dans le sens 1 si on n'appuie pas sur l'interrupteur A0 et à une fréquence de 200 Hz dans le sens 2 si on appuie sur A0. Vous utiliserez l'outil de décision .

## Pilotage d'un servomoteur

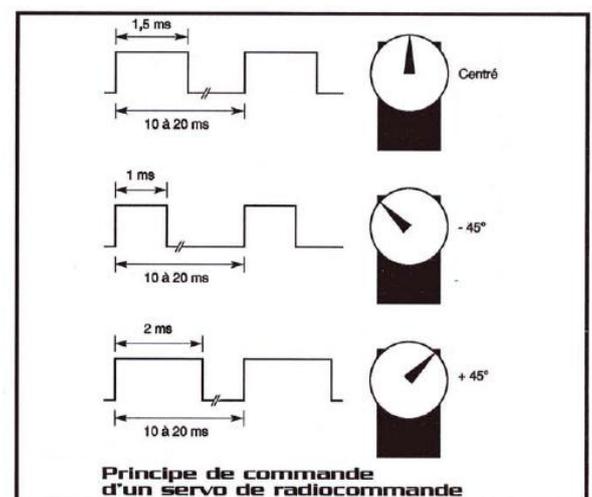
Un servomoteur analogique se présente sous un aspect physique similaire à ce que vous pouvez découvrir ci-dessous



Coté électrique, un tel servomoteur ne dispose que de trois fils codés par des couleurs. Les fils rouge et noir sont destinés à son alimentation qui peut être comprise entre 4.8V et 6V. Le troisième fil de couleur jaune ou blanche sert à transmettre les ordres au servomoteur sous forme d'impulsions codées en largeur.

Principe de codage :

- Une impulsion de 1.5ms de large place le servo dans sa position dite centrée ou de repos.
- Une impulsion de 1ms de large fait tourner le servomoteur dans sa position maximale en sens inverse des aiguilles d'une montre. Ce qui représente un angle de  $45^\circ$  par rapport sa position de repos.
- Une impulsion de 2ms de large fait tourner le servomoteur dans sa position maximale dans le sens des aiguilles d'une montre. Ce qui représente là aussi un angle de  $45^\circ$  par rapport sa position de repos.



- Toute autre largeur d'impulsion comprise entre 1 et 2ms permet d'obtenir les innombrables positions intermédiaires

Ces dernières doivent se répéter à un rythme tel que l'on ne doit pas avoir plus de 10 à 20ms d'écart entre deux impulsions successives.

Une fois que le servomoteur à atteint la position désirée, il n'est plus absolument indispensable de continuer à lui envoyer les impulsions de commande avec la période de répétition de 10 à 20ms.

Si vous le faites, le servomoteur tiendra sa position avec toute la force permise par son moteur et son engrenage, ce qui est évidemment la solution idéale. Si vous ne le faites pas, le servomoteur tiendra la position atteinte que si l'organe commandé ne force pas trop sur son arbre de sortie (dans une telle situation le moteur n'est plus commandé)

Il est alors très simple de programmer un servomoteur. Sous Flowcode, toutes les opérations (calculs, tests, etc....) sont en décimal. Cette valeur est une valeur décimale comprise entre 0 et 255, c'est pourquoi :

- Une durée de 1ms vaudra 0,
- Une durée de 1.5ms vaudra 128,
- Une durée de 2ms vaudra 255.

Afin de piloter un servomoteur sous Flowcode, on utilise l'outil SERVO .

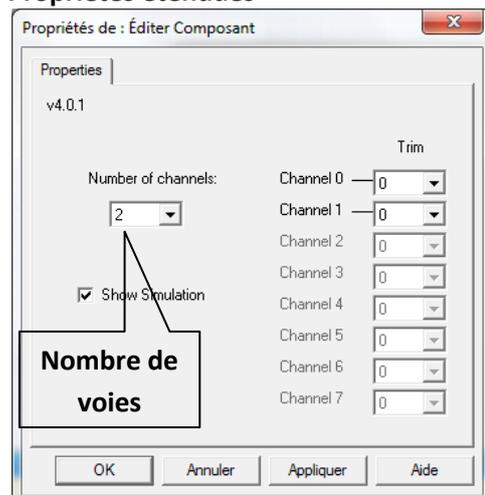


Outil  
servomoteur

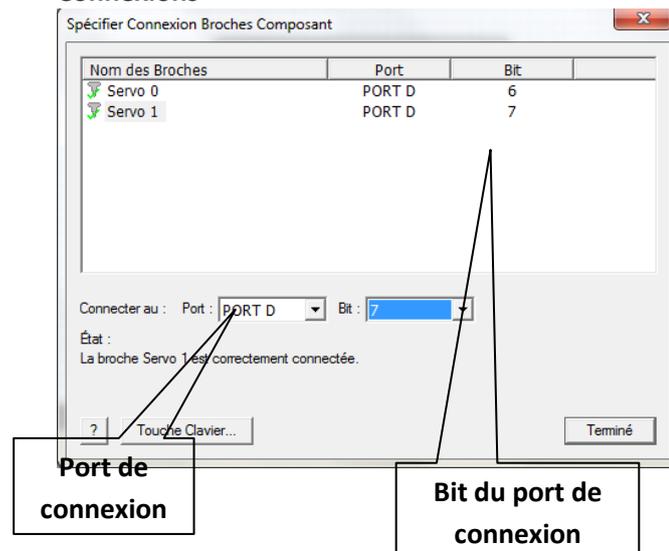


En cliquant droit sur l'objet on peut accéder

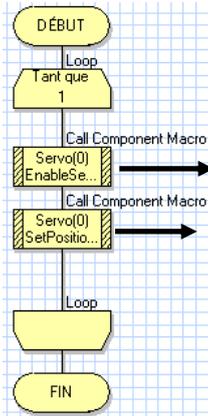
### Propriétés étendues



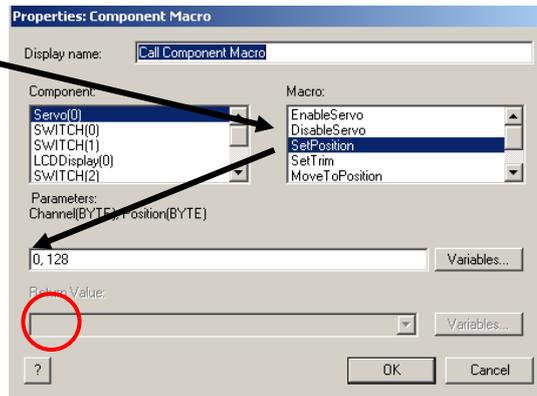
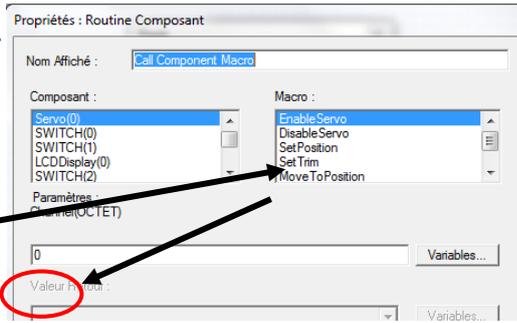
### Connexions



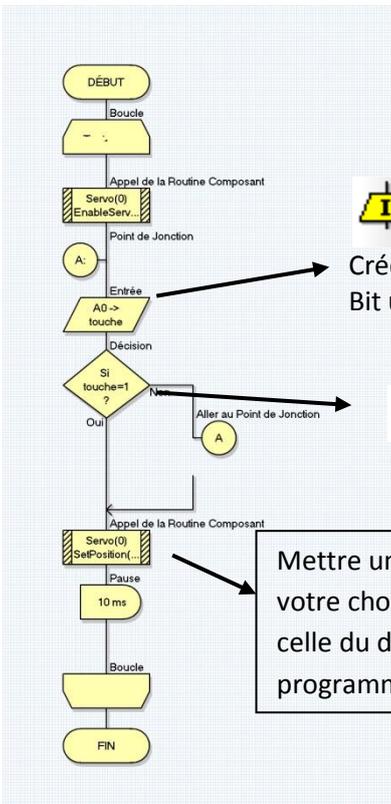
- Réaliser le programme suivant puis simuler



- Sélectionner  (Component Macro)
- Valider le servomoteur de la voie 0 (PORTD bit 6)
- Mettre 0,128 dans Channel, Position



- Sans toucher au servomoteur remplacer 128 par 255. Simuler le programme, que constatez-vous ? Prendre 2 ou 3 valeurs intermédiaire afin de valider le fonctionnement. Conclure, quelle est la course utile du servomoteur ?



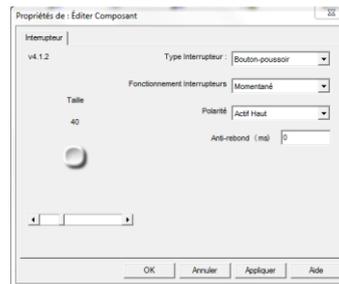
Créer variable touche Bit unique A0

Mettre une valeur de votre choix (différente de celle du dernier programme)

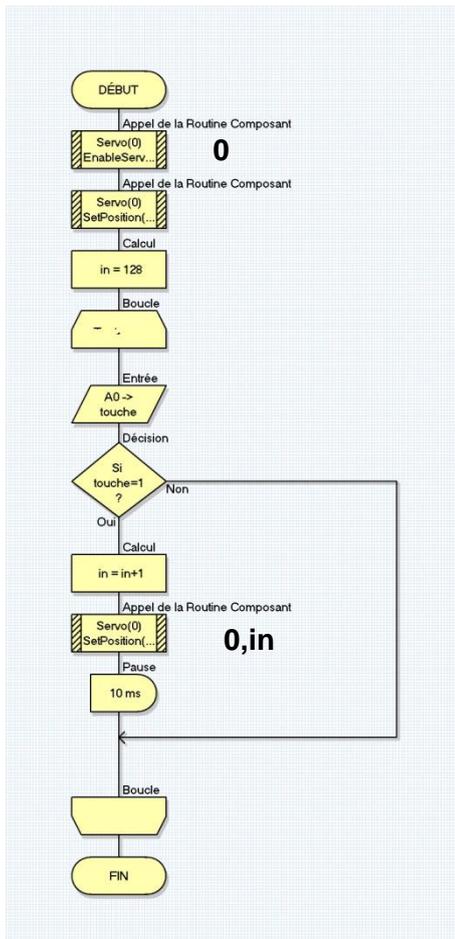
- Réaliser le programme suivant pour le servo0.
- Rajouter un interrupteur, et le relier à A0. Cliquer 'droit' sur l'interrupteur :

- Connexion : 'Port A bit 0'

- Propriétés étendues :



- Simuler le programme. Que se passe t-il ?
- Appuyer sur le bouton poussoir A0, que se passe t-il ?



- Réaliser puis lancer le programme suivant.
- Que fait le servomoteur au démarrage ? Pourquoi ?
- Appuyer plusieurs fois sur le bouton poussoir A0, que se passe t-il ? Justifier.
- Lorsque le servo arrive en bout de course il revient à 0, pourquoi ? proposer une solution pour résoudre le problème.
- Modifier le programme afin de piloter le servomoteur dans l'autre sens

- Réaliser un programme qui permet de faire tourner le servomoteur dans le sens 1 si on appuie sur l'interrupteur A0 et dans le sens 2 si on appuie sur A1. Vous utiliserez l'outil de décision .
- Télécharger  le programme dans le pic et le tester sur la carte de développement.