

Centres d'intérêt abordés	Information
Niveau d'analyse	Comportemental
Objectifs pédagogiques	3.1.4 Traitement de l'information
Connaissances	Traitement programmé : structure à base de microcontrôleur
Activités (3 H)	Analyser et mettre en œuvre la gestion d'entrées-sorties
Ressources documentaires	Cours : Traitement programmé
Ressources matérielles	Ordinateur avec logiciel FLOWCODE

1. PRÉSENTATION

Ce TP d'initiation à la programmation par ordigramme utilise le logiciel FLOWCODE. Il intègre certaines étapes du développement d'un programme :

- traduction de la fonction à réaliser par un ordigramme ;
- test de l'ordigramme par une simulation.

2. DESCRIPTION DES ENTRÉES ET SORTIES UTILISÉES DANS LES PROGRAMMES

Les programmes à écrire sont destinés à être implantés dans un PIC 16F 877A :

- 8 leds sont reliées au PORTD. Elles s'éclairent par un niveau logique '1' ;
- le PORTC reçoit les informations provenant de 8 interrupteurs ;
- un afficheur à cristaux liquides de 2 lignes est relié au PORTB.

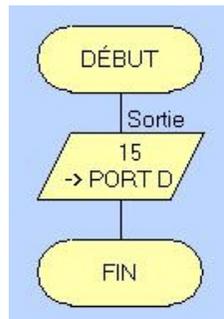
Le schéma structurel simplifié est donné ci-contre.

3. PROGRAMME D'ÉCRITURE SUR UN PORT

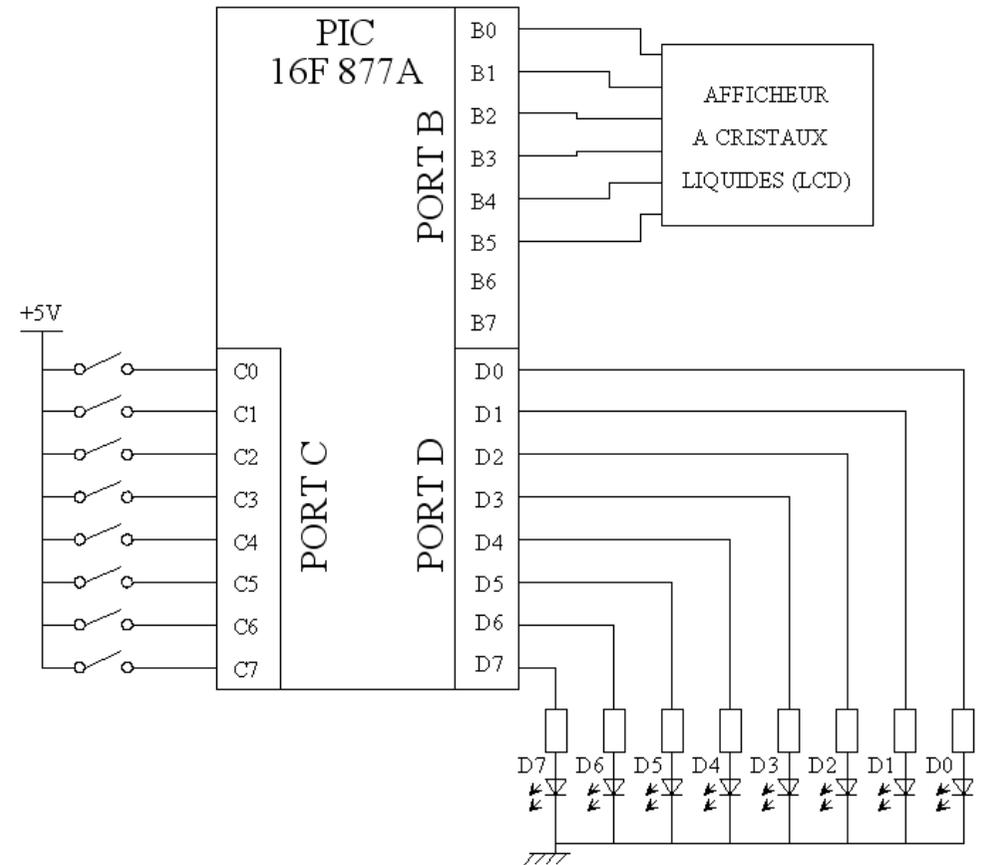
3.1. ANALYSE D'UN PROGRAMME EXISTANT

Le fichier à analyser s'appelle *sortie.fcf*. Il est présent dans le répertoire *documents en consultation* du lecteur de la classe. Avec le logiciel FLOWCODE, ouvrez ce fichier et sauvegardez-le sur votre lecteur personnel.

Simuler le fonctionnement.



Que se passe-t-il ?



Quel symbole est utilisé pour écrire une valeur sur le PORT D :

Quelle valeur faut-il écrire pour que tous les voyants soient allumés ? _____

Quelle valeur faut-il écrire pour que tous les voyants soient éteints ? _____

Vérifier par simulation avec les valeurs trouvées.

FAIRE VÉRIFIER LA SIMULATION PAR LE PROFESSEUR.

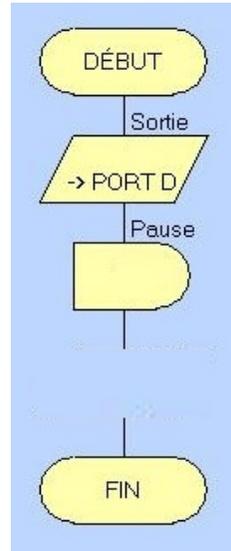
3.2. CLIGNOTANT

☞ Compléter l'ordinogramme pour que tous les voyants restent allumés pendant 1 seconde (ajouter le symbole Pause). Simuler le fonctionnement.

☞ Reproduire ci-contre l'ordinogramme testé :

☞ Compléter l'ordinogramme pour que tous les voyants clignotent (allumés et éteints pendant une seconde). Pour cela il faut ajouter les points de jonction au début et à la fin de l'ordinogramme.

Tester le fonctionnement par simulation.



FAIRE VÉRIFIER LA SIMULATION PAR LE PROFESSEUR.

4. PROGRAMME DE LECTURE D'UN PORT

4.1. ANALYSE D'UN PROGRAMME EXISTANT

Avec le logiciel FLOWCODE, ouvrez le fichier nommé *entree.fcf*.

☞ Simuler le fonctionnement en mode pas par pas.

☞ Relever la valeur de la variable MEMO (en décimal et en hexadécimal) dans les cas suivants :

Position des interrupteurs C7 à C0	Contenu de la variable MEMO		
	Décimal	Hexadécimal	Binaire
tous ouverts			
C3 fermé			
C7 et C5 fermés			
tous fermés			

☞ Convertir les valeurs en binaire.

4.2. RECOPIE DU PORTC SUR LE PORTD

4.2.1. EDITION DE L'ORDINOGRAMME

☞ Compléter l'ordinogramme pour que l'état des interrupteurs soit visualisé sur les leds du PORT D.

4.2.2. SIMULATION DU FONCTIONNEMENT

Placer sur le schéma les leds reliées au PORT D. Tester par simulation le fonctionnement de l'ordinogramme.

5. PROGRAMMES UTILISANT L'AFFICHEUR À CRISTAUX LIQUIDES

5.1. AFFICHAGE D'UN MESSAGE

Pour afficher un caractère ou un message sur l'afficheur, on fait appel à deux macros prédéfinies dédiées à l'afficheur à cristaux liquides (LCDDisplay) :

- la routine composant Init doit être exécutée une seule fois pour configurer et initialiser l'afficheur ;
- la routine composant Ecrit_Chaine(Chaine) permet d'afficher une chaîne de caractères ASCII.

Exemples : Ecrit_Chaine("Bonjour") affichera la chaîne de caractères écrite entre les symboles " ", Ecrit_Car ('B') affichera le caractère écrit entre les symboles ' '.

5.1.1. EDITION DE L'ORDINOGRAMME

On veut écrire un programme qui affiche le message "Bonjour".

☞ Avec le logiciel FLOWCODE, ouvrez le fichier nommé *affiche.fcf*.

Dans l'ordinogramme, placer une routine composant avant la fin. Double-cliquer sur la routine composant. Dans la boîte de dialogue, cliquer sur le bouton Composant. Sélectionner le composant LCDDisplay0 et la routine composant Ecrit_Chaine.

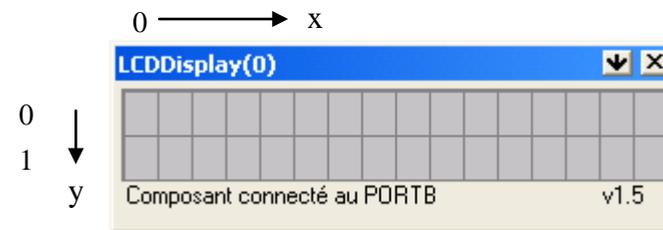
Compléter l'onglet Chaîne de caractères pour afficher le message voulu.

5.1.2. SIMULATION DU FONCTIONNEMENT

Tester par simulation le fonctionnement de l'ordinogramme.

5.2. AFFICHAGE D'UN MESSAGE SUR DEUX LIGNES

Le curseur définit la position d'affichage du prochain caractère. Il est possible de modifier sa position en utilisant la routine composant Curseur(x,y) où x et y sont les coordonnées du curseur.



☞ Modifier l'ordinogramme pour afficher le message "Bonjour" sur la première ligne et la date sur la seconde ligne. Tester le fonctionnement.

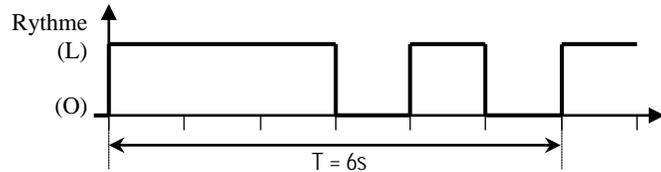
FAIRE VÉRIFIER LA SIMULATION PAR LE PROFESSEUR.

6. APPLICATIONS

6.1. BALISE MARITIME

6.1.1. FONCTIONNEMENT

La balise maritime est équipée à son sommet d'un système d'éclairage qui sert à guider les bateaux dans la nuit. Le signal lumineux émis par la balise est intermittent et possède un rythme propre qui permet de l'identifier. Le rythme est donné par la répartition des temps de lumière (L) et d'obscurité (O) :



Une cellule photoélectrique permet de réaliser la détection du jour et de la nuit : la balise va s'allumer automatiquement la nuit et s'éteindre le jour.

6.1.2. ENTRÉES – SORTIES

La cellule de détection du jour et de la nuit produit une information logique sur C7. C7 = '1' lorsqu'il fait nuit.

On utilisera une variable NUIT pour mémoriser cette information.

La lampe qui s'allume et s'éteint selon le rythme défini sera reliée à D0.

6.1.3. ÉDITION DE L'ORDINOGRAMME

Avec le logiciel FLOWCODE, ouvrez un nouveau fichier. Sélectionner le PIC 16F877A.

☞ Représenter l'ordinogramme qui traduit le fonctionnement automatique de la balise. Enregistrer le fichier sous le nom *balise.fcf*.

6.1.4. SIMULATION DU FONCTIONNEMENT

Placer sur le schéma la led et l'interrupteur. Tester par simulation le fonctionnement de l'ordinogramme.

FAIRE VÉRIFIER LA SIMULATION PAR LE PROFESSEUR.

6.2. RÉALISATION DE LA COMMANDE DU VOLET ROULANT AUTOMATIQUE PAR ORDINOGRAMME

L'objectif de cette partie est d'écrire le programme qui réalise la commande du volet roulant automatique étudié lors du TP N°2.

6.2.1. RAPPELS SUR LE FONCTIONNEMENT

La fonction à programmer possède quatre entrées :

- Auto/Manu (A) : A = 1 en mode automatique, A = 0 en mode manuel (C0) ;
- Jour/Nuit (J) : J = 1 le jour, J = 0 la nuit (C1) ;
- Consigne manuelle de montée (M) : active à 1 (C2) ;
- Consigne manuelle de descente (D) : active à 1 (C3).

L'analyse du fonctionnement a permis d'établir les équations des sorties Sm et Sd :

- $S_m = A \cdot J + \bar{A} \cdot M \cdot \bar{D}$
- $S_d = A \cdot \bar{J} + \bar{A} \cdot D \cdot \bar{M}$

Les sorties Sm et Sd seront connectées respectivement à D0 et D1.

6.2.2. ÉDITION DE L'ORDINOGRAMME

Avec le logiciel FLOWCODE, ouvrez le fichier un nouveau fichier. Sélectionner le PIC 16F877A.

☞ Représenter l'ordinogramme qui traduit le fonctionnement du volet roulant automatique. Enregistrer le fichier sous le nom *volet.fcf*.

6.2.3. SIMULATION DU FONCTIONNEMENT

☞ Placer sur le schéma les leds et les interrupteurs. Pour faciliter la simulation, affecter une étiquette à chacune des entrées et sorties.

Tester par simulation le fonctionnement de l'ordinogramme.

6.2.4. AFFICHAGE DU FONCTIONNEMENT

Compléter l'ordinogramme pour afficher les messages suivants pendant les différentes phases de fonctionnement :

- "MONTEE" pendant la phase de montée.
- "DESCENTE" pendant la phase de descente.

Tester le fonctionnement.

FAIRE VÉRIFIER LA SIMULATION PAR LE PROFESSEUR.

7. ORDINOGRAMMES SUPPLÉMENTAIRES

7.1. DÉFILEMENT

7.1.1. FONCTIONNEMENT

On veut écrire un programme qui allume successivement chacune des 8 leds (B0 à B7) pendant 200ms. Ces opérations sont ensuite répétées indéfiniment.

7.1.2. ÉDITION DE L'ORDINOGRAMME

Avec le logiciel FLOWCODE, ouvrez le fichier un nouveau fichier. Sélectionner le PIC 16F877A.

☞ Représenter l'ordinogramme qui traduit le fonctionnement. On utilisera une variable nommée DEFILE (de type entier) pour mémoriser l'information envoyée vers les leds

Enregistrer le fichier sous le nom *defilement.fcf*.

7.1.3. SIMULATION DU FONCTIONNEMENT

☞ Placer sur le schéma les leds reliées au PORT B.

Tester par simulation le fonctionnement de l'ordinogramme.

FAIRE VÉRIFIER LA SIMULATION PAR LE PROFESSEUR.

7.2. AFFICHAGE

Reprendre l'ordinogramme AFFICHE en le sauvegardant sous le nom AFFICHE2.

Modifier cet ord inogramme pour afficher le message :

- " ouvert" sur la première ligne si l'interrupteur C0 est ouvert ;
- " fermé" sur la seconde ligne si l'interrupteur C0 est fermé.

L'ensemble de ces opérations est réalisé en boucle.

Tester par simulation le fonctionnement de l'ordinogramme.

8. SYNTHÈSE FLOWCODE

☞ Nommer et représenter le symbole qui permet de faire une opération d'écriture sur un PORT :

☞ Nommer et représenter le symbole qui permet de faire une opération de lecture d'un PORT :

☞ Que faut-il créer lors d'une opération de lecture d'un PORT :

☞ Nommer et représenter les symboles qui permettent de faire un rebouclage :

☞ Nommer et représenter le symbole qui permet de faire une opération logique ou arithmétique:

☞ Nommer et représenter le symbole qui permet de faire un test :

☞ Quelle opération doit précéder le test quand on veut tester un bit d'un PORT ?
