

01/01/2013

Programmation Flowcode

- *Prototypage et Débogage Matériel*

MLK - ECST

Programmation Flowcode

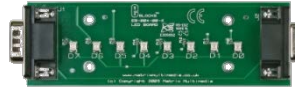
- Prototypage et Débogage Matériel

Centre d'intérêt :	CI4 Structures matérielles et logicielles associées au traitement de l'info
Objectif et Compétences :	CO8.sin1 Rechercher et choisir une structure logicielle ou matérielle au regard de la définition d'un système
Compétences spécifiques :	Structures fondamentales, masquage logiciel, Débogage pas à pas matériel
Pré-requis :	Cours/TP ETC « Traitement programmé »
Type :	TP déductif

Présentation du matériel de prototypage

Les Eblocks® de Matrix sont des petites cartes qui contiennent les fonctions électroniques élémentaires que l'on rencontre dans les systèmes électroniques embarqués. Elles peuvent facilement être interconnectées et ainsi réaliser une multitude de systèmes complexes. Elles forment une très bonne base pour la réalisation de prototypage rapide. Une fois que le système développé est opérationnel, un circuit personnalisé peut rapidement être créé à partir de l'assemblage réalisé.

- Voici une liste non exhaustive des Eblocks® mis à votre disposition :

EB006 - Multiprogrammateur  Elle permet de programmer les PIC à 8, 14, 18, 20, 28 et 40 broches des séries 12, 16 and 18. Elle dispose également de 5 ports pour recevoir les autres E-blocks.	EB004 - Platine à LED  Elle comporte 8 LED ainsi que des connecteurs entrant et sortant. Il est ainsi possible de connecter d'autres E-blocks à ce module.	EB007 - Platine commutateurs  Elle comporte 8 boutons poussoirs actifs à la fermeture ainsi que des connecteurs entrant et sortant. Il est ainsi possible de connecter d'autres E-blocks à ce module.
EB005 - Platine LCD  Elle comporte un afficheur LCD de 2 lignes de 16 caractères interfacé sur un bus série de 5 fils. Des macro-commandes sont disponibles dans Flowcode pour piloter ce module.	EB002 - Platine borniers à vis  Elle dispose de 8 borniers à vis qui peuvent être utilisés pour connecter des périphériques externes à un système composé d'Eblocks®. Ces entrées / sorties sont protégées par des résistances de 150 ohms.	EB011 - Platine puissance  Elle contient des sorties de puissance à usage général pour commander des lampes ou des moteurs – y compris des moteurs pas à pas. Ces sorties sont utilisables jusqu'à 36 V 500 mA et protégées par un fusible réarmable.
EB014 - Platine pavé numérique  Un clavier matricé 4x3 qui permet d'envoyer des données dans vos systèmes. Des macro-commandes sont disponibles dans Flowcode pour piloter ce module.	EB008 - Platine afficheurs 7 segments  Elle comporte quatre afficheurs 7 segments à anodes communes. Des macro-commandes sont disponibles dans Flowcode pour piloter ce module.	EB066 - Platine GSM / GPRS  Elle fournit un accès aux réseaux de téléphonie GSM/GPRS afin d'échanger des appels vocaux ou SMS. Elle inclut une antenne, un emplacement pour carte SIM, des jacks 2.5mm pour casque et micro, ainsi qu'une LED pour connaître l'état du réseau.

Existe également : EB024 - Platine Bluetooth, EB069 - Platine LAN Wireless, EB056 - Platine de géolocalisation GPS, etc.

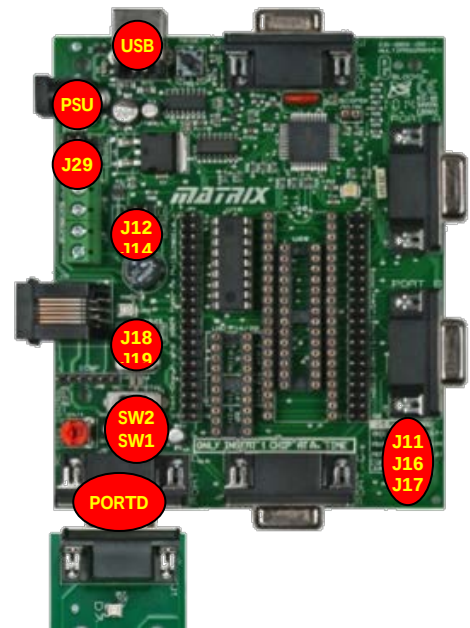
Configuration du matériel

Les notices des Eblocks® sont disponibles à l'adresse suivante : <http://www.multipower.fr/flo1/fblock.html>

- Afin de configurer correctement le matériel, suivre l'extrait de la notice de la carte EB006 Multiprogrammateur :

Testing the board – with an external power supply :

- 1) Ensure power is supplied to the Multiprogrammer board
 1. USB cable required
 2. PSU cable required
 - 2) Set Jumper J29 to 'PSU'
 - 3) Set Jumper J12, J14 to 'USB'
 - 4) Set Jumpers J11, J16 and J17 to 'I/O PORT'
 - 5) Set Jumper J18, J19 to 'OSC'
 - 6) XTAL mode (SW2 towards the EDGE of the board)
 - 7) FAST mode (SW1 towards the CENTRE of the board)
 - 8) Insert EB - 004 LED Board into Port D
- Enfin, vérifier la présence d'un PIC16F877A sur le support 40 broches sur la carte EB006 Multiprogrammateur.



Configuration du logiciel Flowcode V4

Remarque : Pour réaliser l'ensemble de ce TP, et notamment le mode débogage matériel (ICD : In-Circuit Debugging), il est indispensable d'utiliser une version 4 ou supérieure de Flowcode.

- Lancer le logiciel *Flowcode*, créer un *Nouvel Algorithme Flowcode* et choisir un *PIC16F877A*.
- Dans l'onglet *Options Globales* choisir la vitesse d'horloge qui correspond à celle indiquée sur le quartz de la carte EB006 Multiprogrammateur (probablement 19660800Hz).



Programmation

Faire Allumer les LED

- Ajouter le composant virtuel *Outputs* -> *LEDarray* et éditer ses propriétés afin qu'elles correspondent à la configuration matérielle.
- Dans une boucle « *Répéter Toujours* » réaliser un ordiogramme qui allume l'ensemble des LED.
- Enregistrer votre travail dans *P:/Mes devoirs/SIN/allumer-LED.fcf*. Simuler votre solution puis transférer votre ordiogramme vers la cible.
- Commenter chaque élément de votre ordiogramme :

Faire vérifier votre résultat par le professeur

Faire Scintiller les LED

1. Structure linéaire

- Modifier votre ordiogramme afin de faire scintiller l'ensemble des LED grâce à une structure linéaire :
Répéter {Allumer LED ; attendre 30ms ; Éteindre LED ; attendre 300ms}
- Enregistrer votre travail dans *P:/Mes devoirs/SIN/scintiller-lineaire.fcf*. Simuler puis transférer votre ordiogramme vers la cible.

2. Structure alternative

Remarque sur les masques logiciels en Ou Exclusif :

Soit a une variable booléenne. On considère l'expression : $a \oplus 1$

- Si $a = 0$ alors $a \oplus 1 =$ _____
- Si $a = 1$ alors $a \oplus 1 =$ _____

Conclusion : Quelle est la fonction logique réalisée par l'opération $\oplus 1$? _____

Exercice préalable : Clignotement simple

- Modifier votre ordinogramme en utilisant un masque logiciel en XOR sur une variable nommée LED, afin de faire clignoter l'ensemble des LED :

Répéter {Complémenter l'état des LED ; attendre 0,5s}

- Enregistrer votre travail dans *P:/Mes devoirs/SIN/Clignoter-XOR.fcf*. Simuler puis transférer votre ordinogramme vers la cible.

Finalisation

- Modifier votre ordinogramme en utilisant un masque logiciel en XOR sur une variable nommée LED, afin de faire scintiller l'ensemble des LED :

Répéter {Complémenter l'état des LED ; Si les LED sont allumées alors attendre 30ms ; Sinon attendre 0,3s}

- Enregistrer votre travail dans *P:/Mes devoirs/SIN/scintiller-alternative.fcf*. Simuler puis transférer votre ordinogramme vers la cible.
- Commenter chaque élément de votre ordinogramme :

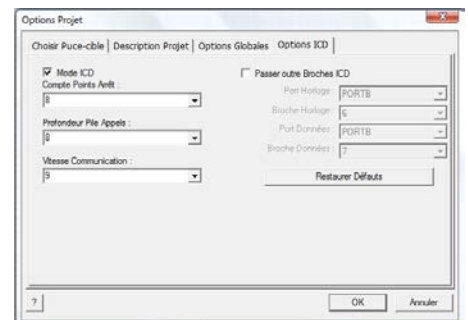
Faire vérifier votre résultat par le professeur

Mode pas à pas logiciel et matériel

Afin de faciliter le débogage d'ordinogrammes, il est parfois utile d'exécuter ces derniers en mode pas à pas. Cela permet de vérifier si les tests ou les sous-programmes (macros) sont correctement exécutés.

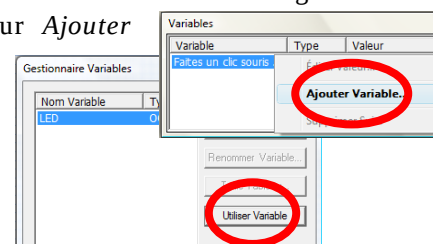
Flowcode permet de faire du pas à pas logiciel en mode simulation et du pas à pas matériel directement sur les cartes électroniques :

- Pour lancer un ordinogramme en mode pas à pas logiciel il suffit de cliquer sur l'une des deux icônes « *mode pas à pas détaillé (F8)* » ou « *mode pas à pas principal (Shift + F8)* »  : Exécuter votre ordinogramme en mode pas à pas et observer les effets de ce dernier sur les composants virtuels.
- Pour lancer un ordinogramme en mode pas à pas matériel il faut préalablement configurer Flowcode :
 - Dans le menu *Build -> Options Projet* et dans l'onglet *Options ICD*, cliquer sur *Mode ICD*.
 - Transférer votre ordinogramme vers la cible, observer qu'il ne s'exécute pas automatiquement sur les cartes.
 - Cliquer sur l'une des deux icônes « *mode pas à pas détaillé (F8)* » ou « *mode pas à pas principal (Shift + F8)* »  : Observer les effets de l'ordinogramme sur les cartes Ebblocks®



Pendant le mode pas à pas, il est possible d'observer l'évolution des variables utilisées dans l'ordinogramme :

- Effectuer un click droit dans la fenêtre *Variables* puis cliquer sur *Ajouter Variable...*
- Sélectionner la ou les variables de votre ordinogramme. Les valeurs de ces dernières seront rafraichies lors de l'exécution de chaque pas. Cela peut s'avérer être une aide précieuse pour le débogage d'applications.
- Lancer votre ordinogramme en mode pas à pas et observer l'évolution de la variable. **Faire vérifier votre résultat par le professeur**



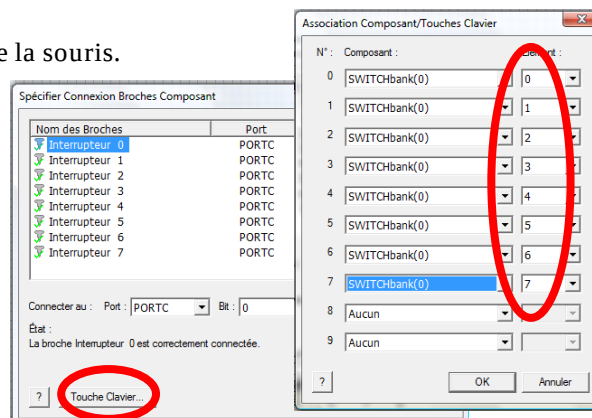
Lecture et écriture



- Câbler ou vérifier la présence d'une carte *EB004 – Platine à LED* et d'une carte *EB007 – Platine commutateurs* sur la carte *EB006 – Multiprogrammateur*. Vérifier également la présence d'un fil d'alimentation +V entre ces deux dernières.
- Créer un nouvel ordinogramme dans votre dossier de travail *P:/Mes devoirs/SIN/lecture-ecriture.fcf*
- Ajouter les composants virtuels *LED Outputs -> LEDarray* et boutons poussoirs *Inputs -> SWITCHbank*. Vérifier leurs propriétés de connexions afin qu'ils correspondent à la configuration matérielle.
- Réaliser un ordinogramme qui lit l'état des commutateurs dans une variable nommée *LED* et qui écrit le résultat sur les voyants connectés au *PORTD*.
- Simuler votre résultat, appuyer sur les boutons à l'aide de la souris.

Remarque : Avec les boutons poussoirs virtuels et la souris, il est impossible de cliquer sur plusieurs boutons en même temps. Il existe cependant une solution (autre que de mettre des interrupteurs à levier) qui consiste à associer une touche numérique de clavier à chaque bouton :

- Dans les propriétés de *Connexions* des poussoirs, cliquer sur *Touche Clavier...*
- Associer chaque élément bouton poussoir à un numéro de touche du clavier.
- Simuler votre résultat, appuyer sur plusieurs touches numériques du clavier d'ordinateur en même temps.
- Transférer votre programme sur la cible en mode ICD puis exécuter le en pas à pas. Observer l'évolution de la Variable *LED* lorsque vous appuyer sur les boutons de la carte *EB007 – Platine commutateurs*.



Faire vérifier votre résultat par le professeur

Défilement

1. De gauche à droite en Structure linéaire

- Créer un nouvel ordinogramme dans votre dossier de travail *P:/Mes devoirs/SIN/defilement-lineaire.fcf* avec les mêmes composants que précédemment (il est possible d'utiliser la commande *Enregistrer sous...*) qui réalise un défilement des LED de gauche à droite et utilisant une structure linéaire :
Répéter {Allumer D7 ; Attendre 50ms ; Allumer D6 ; Attendre 50ms ; ... ; Allumer D0 ; Attendre 50ms}
- Tester votre solution par simulation et sur la cible matérielle.

2. De gauche à droite en Structure répétitive « Faire... Tant que »

- Créer un nouvel ordinogramme dans votre dossier de travail *P:/Mes devoirs/SIN/defilement-repetitive.fcf* avec les mêmes composants que précédemment qui réalise un défilement des voyants de gauche à droite et utilisant une structure répétitive « Faire... Tant que ». On utilisera une variable nommée *LED* qu'on initialisera, transférera sur le *PORTD* puis décalera à droite avant chaque « rebouclage ».
- Déboguer votre solution sur la cible matérielle en mode ICD et en surveillant la variable *LED*.
- Commenter chaque élément de votre ordinogramme : **Faire vérifier votre résultat par le professeur**

3. Arrêt du défilement

- Modifier votre ordinogramme de manière à ce que le défilement s'arrête lorsque les boutons poussoirs SW3 et SW4 sont enfoncés simultanément. On utilisera une variable nommée BOUTONS pour lire l'état des switchs.
- Débuguer votre solution sur la cible matérielle en mode ICD et en surveillant les variables LED et BOUTONS.
- Commenter les éléments qui diffèrent de la version précédente :

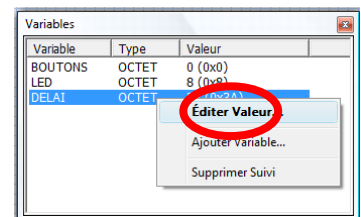
Faire vérifier votre résultat par le professeur

4. Défilement décéléré

- Modifier votre ordinogramme en remplaçant la constante de 50ms dans la fonction *Pause* par une variable nommée DELAI. Cette variable démarrera à 0 puis s'incrémentera de 1 à chaque défilement complet des 8 voyants.
- Tester votre solution sur la cible matérielle.

Remarque : La décélération est nettement visible lorsque la variable DELAI est proche de 0 puis, plus cette dernière augmente et plus la différence de vitesse est négligeable. Le « rebouclage » de 255 à 0 (sur un octet) peut sembler très long. Il est possible de modifier la valeur de la variable DELAI lors de l'exécution de l'ordinogramme en mode ICD :

- Exécuter votre solution sur la cible matérielle en mode ICD (transférer sur la cible  puis lancer l'ordinogramme ). Cliquer sur *Suspendre la simulation*  et demander la surveillance de la variable DELAI.
- Exécuter l'ordinogramme en mode pas à pas sur plusieurs cycles afin de voir s'incrémenter la variable DELAI.
- Effectuer un click droit sur la variable DELAI puis cliquer sur *Éditer Valeur*.
- Entrer une valeur proche de 255 (par exemple 250) puis relancer l'ordinogramme .



Faire vérifier votre résultat par le professeur

5. Défilement de gauche à droite puis de droite à gauche

- Modifier votre ordinogramme pour que les voyantes défilent dans les deux sens. Votre solution devra conserver les fonctionnalités précédentes : Arrêt du défilement et Défilement décéléré.

Astuce : Copier l'ensemble de la boucle « Faire... Tant que » qui opère le décalage de gauche à droite et coller la une seconde fois à la suite. Il ne reste plus qu'à ajuster les conditions initiales et le sens de décalage.

- Débuguer votre solution sur la cible matérielle en mode ICD et en surveillant les variables.

Faire vérifier votre résultat par le professeur

6. Défilement de gauche à droite et de droite à gauche simultané

Remarque sur les masques logiciels en OU :

Soit a une variable booléenne :

On considère l'expression : $a + 0$

- Si $a = 0$ alors $a + 0 =$ _____
- Si $a = 1$ alors $a + 0 =$ _____

Conclusion : Quelle est la fonction logique réalisée par l'opération $+0$? _____

Soit a une variable booléenne :

On considère l'expression : $a + 1$

- Si $a = 0$ alors $a + 1 =$ _____
- Si $a = 1$ alors $a + 1 =$ _____

Conclusion : Quelle est la fonction logique réalisée par l'opération $+1$? _____

- Sauvegarder votre précédent travail puis créer un nouvel ordiogramme avec la commande *Enregistrer sous...* P:/Mes devoirs/SIN/defilement-simultane.fcf. Ce dernier doit réaliser un défilement simultané des voyants de gauche à droite et de droite à gauche (avec deux voyants en même temps). Pour cela, utiliser :
 - Une variable LED1 pour le défilement à droite,
 - Une variable LED2 pour le défilement à gauche,
 - La variable LED recevra le résultat du masquage logiciel en OR de LED1 et LED2.
- Débuguer votre solution sur la cible matérielle en mode ICD et en surveillant les variables.

Faire vérifier votre résultat par le professeur

7. Défilement gauche à droite et droite à gauche simultané de LED éteintes

Cette fois ce ne sont pas des voyants allumés qui doivent défiler mais des voyants éteints.

Remarque sur les masques logiciels en ET :

Soit a une variable booléenne :

On considère l'expression : $a \cdot 0$

- Si $a = 0$ alors $a \cdot 0 =$ _____
- Si $a = 1$ alors $a \cdot 0 =$ _____

Conclusion : Quelle est la fonction logique réalisée par l'opération $\cdot 0$? _____

Soit a une variable booléenne :

On considère l'expression : $a \cdot 1$

- Si $a = 0$ alors $a \cdot 1 =$ _____
- Si $a = 1$ alors $a \cdot 1 =$ _____

Conclusion : Quelle est la fonction logique réalisée par l'opération $\cdot 1$? _____

- Modifier votre ordiogramme pour que les voyants éteints défilent simultanément dans les deux sens. Utiliser pour cela les masques logiciels en AND.

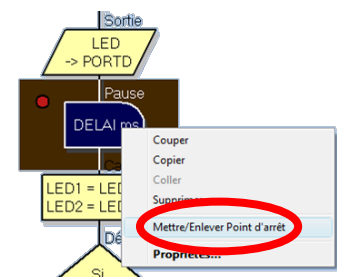
Remarque : les opérations de décalage sous Flowcode introduisent un 0 logique, il faudra donc forcer l'introduction d'un 1 logique.

- Débuguer votre solution sur la cible matérielle en mode ICD et en surveillant les variables.

Débugage avec Points d'arrêts

Il est parfois difficile d'exécuter un ordiogramme complet en mode pas à pas lorsque ce dernier est très long et il s'avère utile d'utiliser des points d'arrêt :

- Effectuer un clic droit sur l'opération qui succède l'écriture de la variable LED sur le PORTD et cliquer sur *Mettre/Enlever Point d'arrêt*
- Exécuter votre solution sur la cible matérielle en mode ICD en cliquant plusieurs fois sur . L'ordiogramme s'arrête automatiquement au point d'arrêt. **Faire vérifier votre résultat par le professeur**



Synthèse

Les masques logicielles

Les masques sont utilisés en informatique pour forcer un ou plusieurs bits d'un mot binaire à un état désiré (0 ou 1). On utilise, pour effectuer des masques, les propriétés de la logique de Boole à savoir : l'élément absorbant et l'élément neutre du OU, du ET et du OU EXCLUSIF.

1. Masque en ET pour forcer un bit à 0

Soit l'état d'une variable = 0b10010111. On veut forcer le bit 1 à '0' sans modifier les autres bits.

Propriétés utilisées : _____

Variable : 1 0 0 1 0 1 1 1	Équivalent en syntaxe Flowcode :
Masque : .	
Résultat : _____	

2. Masque en OU pour forcer un bit à 1

Soit l'état d'une variable = 0b10010111. On veut forcer le bit 3 à '1' sans modifier les autres bits.

Propriétés utilisées : _____

Variable : 1 0 0 1 0 1 1 1	Équivalent en syntaxe Flowcode :
Masque : +	
Résultat : _____	

3. Masque en OU EXCLUSIF pour faire basculer l'état d'un bit

Soit l'état d'une variable = 0b100X0111. On veut faire basculer l'état du bit 4 sans modifier les autres bits.

Propriétés utilisées : _____

Variable : 1 0 0 X 0 1 1 1	Équivalent syntaxe Flowcode :
Masque : ⊕	
Résultat : _____	

4. Exercice d'application en syntaxe Flowcode

- Forcer le bit 5 à '0' et le bit 2 à '1' d'une variable :

Variable = _____

- Forcer le bit 7 à '1' et les bits 0 à 3 à '0' d'une variable :

Variable = _____

- Faire basculer les bits 2 à 6 d'une variable :

Variable = _____