

Manuel de FLOWCODE

I) INTRODUCTION

PIC et PIC et ...



1) Présentation de flowcode :

Il utilise le standard international des symboles d'organigrammes ISO5807.

Il permet d'enfourer du code C ou de l'assembleur sous forme de macro.

Il supporte :

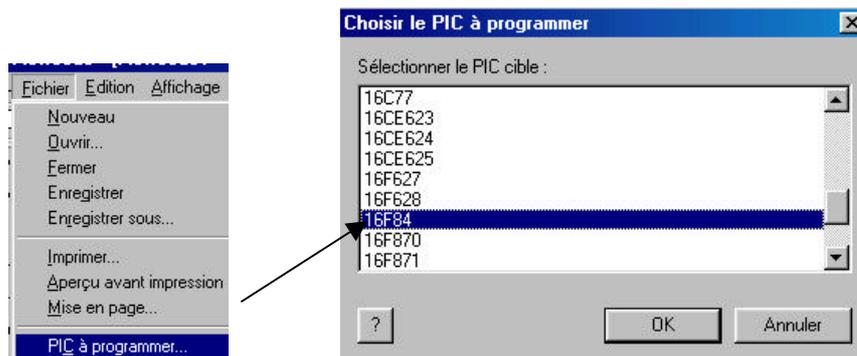
- le paramétrage du Timer 0,
- le paramétrage des convertisseurs A/N 10 bits,
- le paramétrage des interruptions INT, PORTB4-B7 et Timer 0.

2) Les processeurs reconnus par Flowcode sont:

16C62,16C620,16C621,16C622,16C63,16C64,16C65,
16C65A,16C66,16C67,16C71,16C710,16C712,16C716,
16C72,16C73,16C74,16C76,16C77,16CE623,16CE624,
16CE625,16C627,16C628,16F84,16F870,16F871,16F872,
16F873,16F874,16F876,16F877.

3) Choix de la cible PIC pour développer votre algorithme

La première étape consiste à choisir un microcontrôleur PIC parmi la liste disponible ci-dessous en fonction des besoins de l'application.





1 Création de variables et calculs avec des variables

Les variables utilisées par flowcode sont des octets positifs uniquement.

5) Opérez un cliqué-glissé pour insérer une formule de calcul

6) Double cliquez sur le symbole pour rédiger l'équation

Pour créer la variable « ALARME »

- 1) Cliquez sur **Afficher Variables...**
- 2) Cliquez sur **Ajouter Variable...**
- 3) Puis sur **Utiliser Variable**
- 4) Initialisez la variable en écrivant : **ALARME=0**

2 Création de l'algorithme : utilisation des entrées/sorties

Texte explicatif

Sélection de la variable dont le contenu sera écrit sur le port sélectionné.

Choix du port en entrée

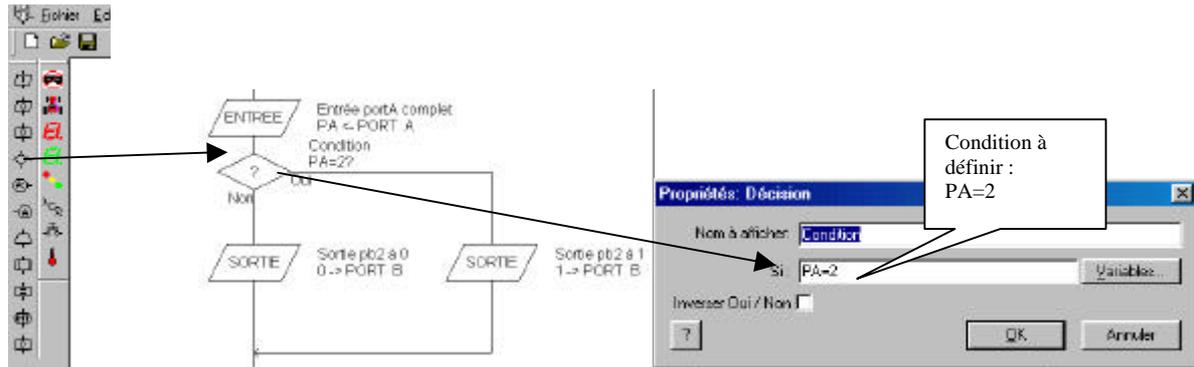
Lecture de 1 bit

Lecture d'un groupe de bit par masquage

Le travail est identique pour un port en sortie. Il est cependant possible d'affecter directement une valeur à un port.



1 Structure alternative complète



Liste des opérateurs conditionnels

()	- Parenthèses.
= < >	- Egal à - Différent de.
+, -, *, / MOD	- Addition, Soustraction, Multiplication, Division - Modulo.
<, <=, >, >=	- Plus petit que, Plus petit ou égal à, Plus grand que, Plus grand ou égal à.
>>, <<	- Décalage à droite, décalage à gauche
NOT, AND, OR, XOR	- NON, ET, OU, OU Exclusif



1 Structure itérative Tant que ... faire

Ecriture de la condition et dans l'exemple : 1

Il est possible d'utiliser des expressions conditionnelles : $A \neq 2, C = 0$

Propriétés : Boucle

Nom à afficher : Boucle infinie

Tant que : 1

Tester la boucle :

au Début

à la Fin

Possibilité de mettre le test avant ou après le traitement

Liste des opérateurs conditionnels

()	- Parenthèses.
= < >	- Egal à - Différent de.
+, -, *, / MOD	- Addition, Soustraction, Multiplication, Division - Modulo.
<, <=, >, >=	- Plus petit que, Plus petit ou égal à, Plus grand que, Plus grand ou égal à.
>>, <<	- Décalage à droite, décalage à gauche
NOT, AND, OR, XOR	- NON, ET, OU, OU Exclusif



1 Les interruptions

Démassage de l'interruption

Source de l'interruption :

- Débordement TIMER 0
- Front montant sur RB0/INT
- Changement d'état sur PB4-PB7

Nom de la macro d'interruption qu'il faudra utiliser pour effectuer le traitement l'interruption sélectionnée.

Pour rédiger votre macro d'IT, il faut ensuite aller dans le menu MACRO puis Edition/Suppression, sélectionner la macro déjà créée par FLOWCODE qui correspond à votre interruption et enfin l'éditer.

Editer/Supprimer une Macro

Choisir la macro :

- INTERRUPT_PORTB
- INTERRUPT_RB0INT
- INTERRUPT_TMRO
- MIS_1_PA
- MISE_1_PB
- RAZ_PA

Editer Supprimer Terminer

Manuel de FLOWCODE

VI) STRUCTURE GLOBALE D'UN ALGORIGRAMME

LES MACROS

PIC et PIC et ...



1 Les macros utilisateurs

Une macro a pour objectif de regrouper dans un même bloc une partie de l'algorithme. Elles seront donc très utilisées pour améliorer la lisibilité des algorithmes et surtout de l'algorithme principal.

Création d'une nouvelle macro : Choisir l'option Nouvelle depuis le menu Macro . Flowcode vous demandera un nom pour cette nouvelle macro.

Cliquez sur OK&Editer pour rédiger votre macros

The image shows a flowchart with a decision diamond 'OC=128?' and two 'Appel d'une Macro' blocks labeled 'FEUX1' and 'ORANGE_CLIGN'. A callout box points to the 'Nouvelle Macro...' button in the 'Macro' menu. Another callout box points to the 'OK & Editer' button. Below, a screenshot of the 'Flowcode - Macro - (ORANGE_CLIGN)' window shows a sequence of steps: DEBUT, Calcul TEST_CUSN = TEST_CUSN.A., SORTIE TEST_CUSN -> PORT B, SORTIE TEST_CUSN -> PORT A, inc séquence feux : COMPT1 = 1, and FIN.

2 Les macros prédéfinies

2.a) Deux types de macros prédéfinies

Les macros prédéfinies par FLOWCODE permettent :

- d'utiliser des interruptions, le nom de la macro est défini par FLOWCODE, mais vous pouvez en rédiger le contenu .
- d'utiliser les composants complexes à savoir l'afficheur à led 1 segment , 4 segments , l'afficheur LCD , le générateur de tension analogique continu 0-5V . Cependant ces 4 macros sont uniquement paramétrables vous ne pouvez donc pas modifier les algorithmes de ces macros.

Attention il faut d'abord créer le composant complexe avant de pouvoir paramétrer la macro correspondante.

Etape 1
Création du composant

Etape 2
Insertion de la macro

Etape 3
Choix d'une macro prédéfinie

Etape 5
Paramétrage de la macro

The image shows a flowchart with a 'Appel d'une Macro' block. A callout box points to the 'Appel d'une Macro' block. Another callout box points to the 'Propriétés: Macro' dialog box. The dialog box shows 'Nom à afficher: Appel d'une Macro', 'Appelle une macro existante', 'Paramétrage: LEDx0, Switched, LED7s=00', 'Macro: Afficheur', 'Paramètres: Digit, Valeur, PointDécimale', and 'Valeur: 1,1,0'. The 'Afficheur' macro is selected in the 'Macro' list.

Manuel de FLOWCODE

VI) STRUCTURE GLOBALE D'UN ALGORIGRAMME

LES MACROS

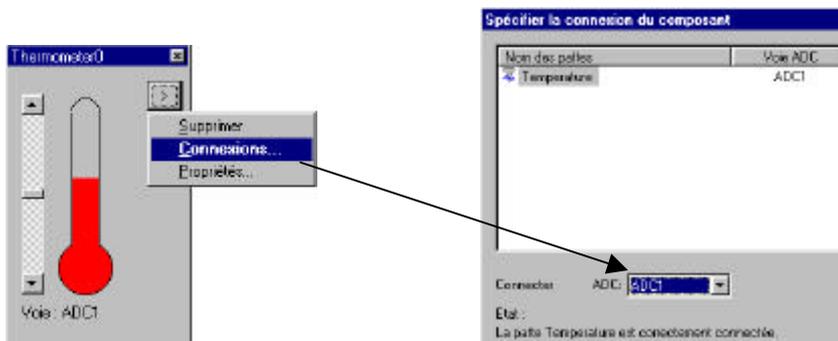
PIC et PIC et ...



2.b) Exemple d'utilisation d'une macro prédéfinie pour gérer le CAN

Pour pouvoir utiliser le CAN, il faut :

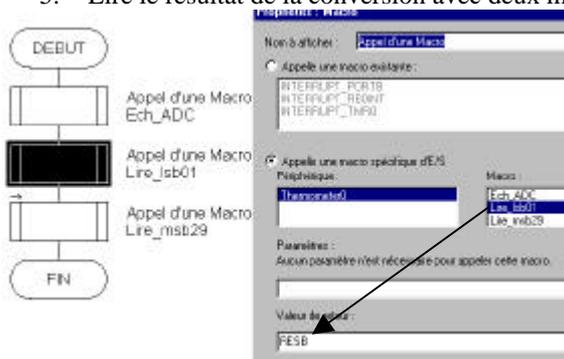
1. Choisir un PIC qui soit équipé du périphérique C.A.N.
2. Cliquer sur  pour activer le composant « générateur de tension analogique continue 0-5V ».
3. Choisir la broche sur laquelle se fera la conversion.



4. Lancer une conversion



5. Lire le résultat de la conversion avec deux macros «Lire_lsb01 » et « Lire_msb29 »



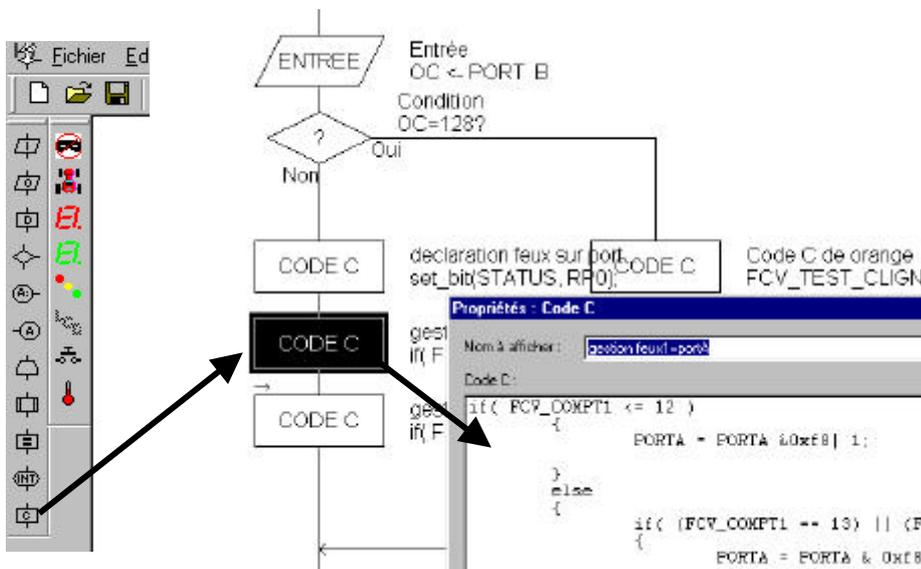
Lire_lsb01
Lit les deux bits de poids faible de la valeur sur 10 bits et les stocke dans une variable.

Lire_msb29
Lit les 8 bits de poids fort de la valeur sur 10 bits et les stocke dans une variable.



3 Insertion de code C ou code Assembleur

Il est possible d'enfourer du code C ou assembleur dans des programmes Flowcode. **Ce code ne pourra pas être simulé par Flowcode**, le test (debuggage) de l'algorigramme sera donc plus difficile.



Remarque sur la visibilité des macros et des variables entre l'algorigramme, le C et l'Assembleur :

- **Algorigramme / Code C**

Pour accéder aux variables Flowcode (ex : PA) , il faut rajouter FCV_ (ex : FCV_PA)

Pour accéder aux macros Flowcode (ex : FEUX1) , il faut rajouter FCM_ (ex : FCM_FEUX1())

- **Algorigramme / Code Assembleur**

Il est possible d'entrer des instructions assembleur dans la fenêtre de Propriétés du code C.
asm

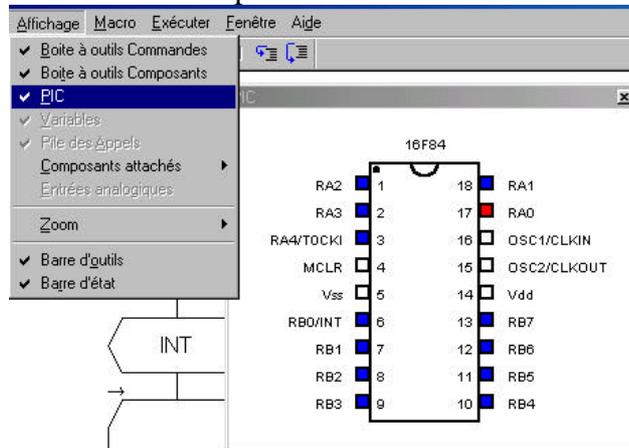
```
{
    ; Entrer vos lignes d'assembleur ici
}
```

Pour accéder aux variables Flowcode (ex :PA) , il faut rajouter _FCV_ (ex : _FCV_PA)

Pour accéder aux macros Flowcode (ex :FEUX1) , il faut rajouter _FCM_ (ex : _FCM_FEUX1)



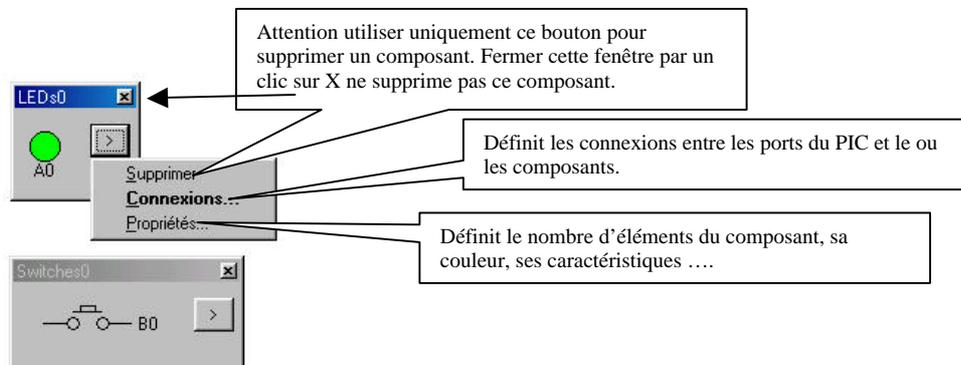
1 Visualisation du composant PIC



2 Utilisation des composants en entrée et sortie

a. Les composants les plus simples

Les composants les plus simples sont les leds , et boutons poussoirs/interrupteurs . Ils peuvent être paramétrés en fonction de l'application.



b. Les composants les plus complexes

Ils sont au nombre de 4 : l'afficheur à led 1 segment , 4 segments , l'afficheur LCD , le générateur de tension analogique continu 0-5V .

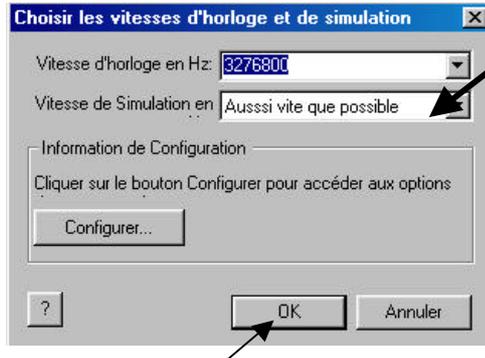
A chacun de ces composants est associée une macro qui doit être paramétrée (cf Manuel de FLOWCODE VI) STRUCTURE GLOBALE D'UN ALGORIGRAMME 2) LES MACROS PREDEFINIES).



3 Simulation de l'algorithme

a) Configuration de la vitesse de simulation

Il faut d'abord configurer la vitesse de simulation dans le menu Exécuter / Vitesse d'horloge. Sélectionnez la vitesse de simulation « aussi vite que possible ».



b) Lancement/ suspension/ arrêt de la simulation

Pour **lancer la simulation** d'un organigramme, sélectionner l'option Exécuter/Continuer en cliquant sur l'icône Exécuter

Cette commande permet **pas à pas**, action après action de simuler l'algorithme **sauf les macros qui sont exécutées d'un bloc**. (Raccourci clavier MAJ+ F8).

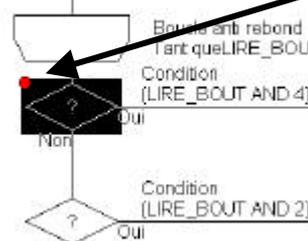
Sélectionner **Pause** dans la simulation pour pouvoir vérifier la valeur des variables

Cette commande permet **pas à pas**, action après action de **simuler l'algorithme**. Elle est idéale pour mieux vérifier le fonctionnement de votre algorithme et suivre la valeur de vos variables. Elle permet de simuler toutes les macros. (raccourci clavier F8).

Arrêt de la simulation

c) Points d'arrêt

Il est possible de définir des points d'arrêt qui vont permettre de stopper le déroulement de la simulation pour visualiser des variables puis de reprendre l'exécution de l'algorithme Avant de lancer la simulation, il faut d'abord sélectionner l'élément de l'algorithme sur lequel doit s'arrêter la simulation puis appuyer sur la touche F9.



Pour relancer le programme il suffit de cliquer sur l'icône Exécuter



1 Paramétrage du mot de configuration du PIC : exemple du 16F870

Au fichier téléchargeable qui contient les instructions en langage machine, il faut ajouter un mot de configuration qui va paramétrer le mode de fonctionnement du microcontrôleur. Ce mot de configuration précise en particulier le type d'horloge, la gamme de fréquence de l'horloge, l'utilisation d'un watchdog, la protection du programme....

Elaboration du mot de configuration en sélectionnant des cases

Fréquence du quartz

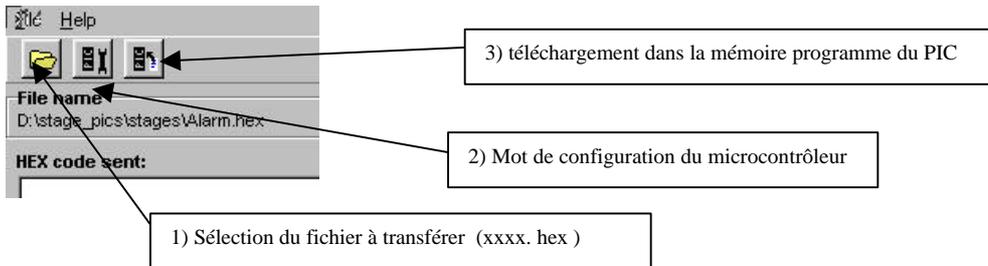
Création automatique du mot de configuration à partir des cases sélectionnées

2 Compilation et téléchargement vers PIC

Lancer la compilation/assemblage/téléchargement par la commande « compiler vers PIC »

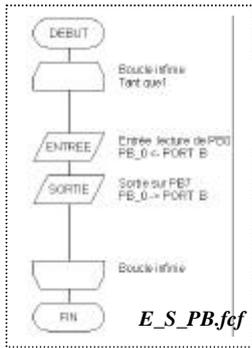


1 Utilisation de PPP



La chaîne de développement, ou implantation de votre algorithme dans la mémoire programme du PIC

PIC et PIC et ...



FLOWCODE
.....
xxxx.fcf

xxxx.c

C2C
.....

xxxx.asm

MPASM
.....

C:/Program Files/mplab/p16f84.inc

xxxx.cod xxxx.err xxxx.lst

xxxx.hex

PPP
.....

Mot de configuration du PIC de 14 bits

Les bits sont envoyés en série dans la mémoire du pic

PIC

PC

Carte de développement PIC

```
.....
char FCV_PB_0;
void main()
{
  set_bit (STATUS, RP0);
  OPTION_REG = 0xC0;
  clear_bit (STATUS, RP0);
  while( 1 )
  {
    ....
    FCV_PB_0 = (PORTB & 0x01 == 0x01);
    ....
    if (FCV_PB_0)
      PORTB = PORTB & 0x7f | 0x80;
    else
      PORTB = PORTB & 0x7f;
  }
  mainendloop: goto mainendloop;
}
E_S_PB.c
```

```
include "C:\PROGRA-1\MPLAB\P16F84.INC"
;Variables *****
_FCV_PB_0      equ 0x0c
ORG 0
goto start__code
start__code
_main__code
bsf STATUS, RP0
movlw D'192'
movwf OPTION_REG
bcf STATUS, RP0
label_0000
bsf STATUS, RP0
movwf TRISB, W
iorlw D'1'
movwf TRISB
bcf STATUS, RP0
movf PORTB, W
andlw D'1'
sublw D'1'
movlw 1
btfss STATUS, Z
clrw
movwf _FCV_PB_0
bsf STATUS, RP0
movf TRISB, W
.....
E_S_PB.asm
```

```
:020000000628D0
:080008000B1109008316C03042
:1000100081008312831606080138860083120608C1
:100020000139013C0130031D03018C0083160608D1
:100030007F39860083120C080319232806087F39AC
:10004000803886000A2806087F3986000A28272873
:00000001FF
E_S_PB.hex
```