

Sciences Industrielles de l'Ingénieur

CI2 – Système à Evènements Discrets (SED)

Partie 2 – Modéliser le comportement d'un SED à l'aide d'un algorigramme

Je dois être capable de :

- Savoir rechercher et reconnaître les principales caractéristiques d'un micro-contrôleur
- Connaître les formes graphiques des structures élémentaires d'un algorigramme

- 1. Les *micro-processeurs* (et notamment les PIC)**
- 2. Les *instructions de base* (principaux blocs)**
- 3. Appel de *fonctions* ou *macros***
- 4. Les *structures décisionnelles***
- 5. Les *structures de boucle***
- 6. Les *fonctions de port d'E/S***

Cl2 – SED, partie 2

1. Les micro-processeurs (et notamment les PIC)

3

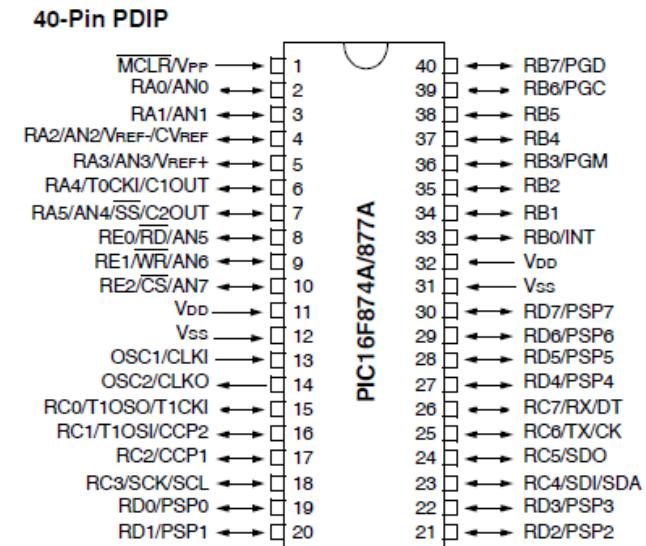
Les microprocesseurs : circuit intégré VLSI (Very Large Scale Intégration) composé de millions de transistors. Nécessite un programme pour fonctionner qui est contenu dans une mémoire externe.

Les microcontrôleurs PIC : circuit intégré plus complexe, qui contient le microprocesseur **et** aussi la mémoire, plus un ensemble de circuits périphériques

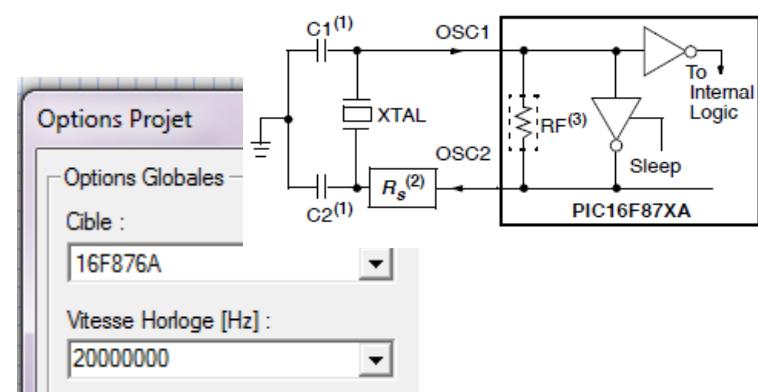
Le programme consiste en une suite d'action en bouclage infini :

- Lecture d'instruction à exécuter
- Décodage de cette instruction
- Exécution de cette instruction

Horloge : La durée d'exécution d'un programme, et donc d'une instruction, dépend directement d'une horloge. On parle de « temps de cycle » qui peuvent varier de quelques ns à µs suivant la fréquence d'oscillation d'un quartz. Dans l'exemple ci-contre, le temps de cycle est de $(20\text{Mhz}) \times 4 = 200\text{ns}$



Le microcontrôleur PIC16F876A



Un quartz (XTAL) de 20Mhz connecté au PIC et paramétré sous FLOWCODE

1. Les micro-processeurs (et notamment les PIC)

Description sommaire de quelques exemples de PIC :

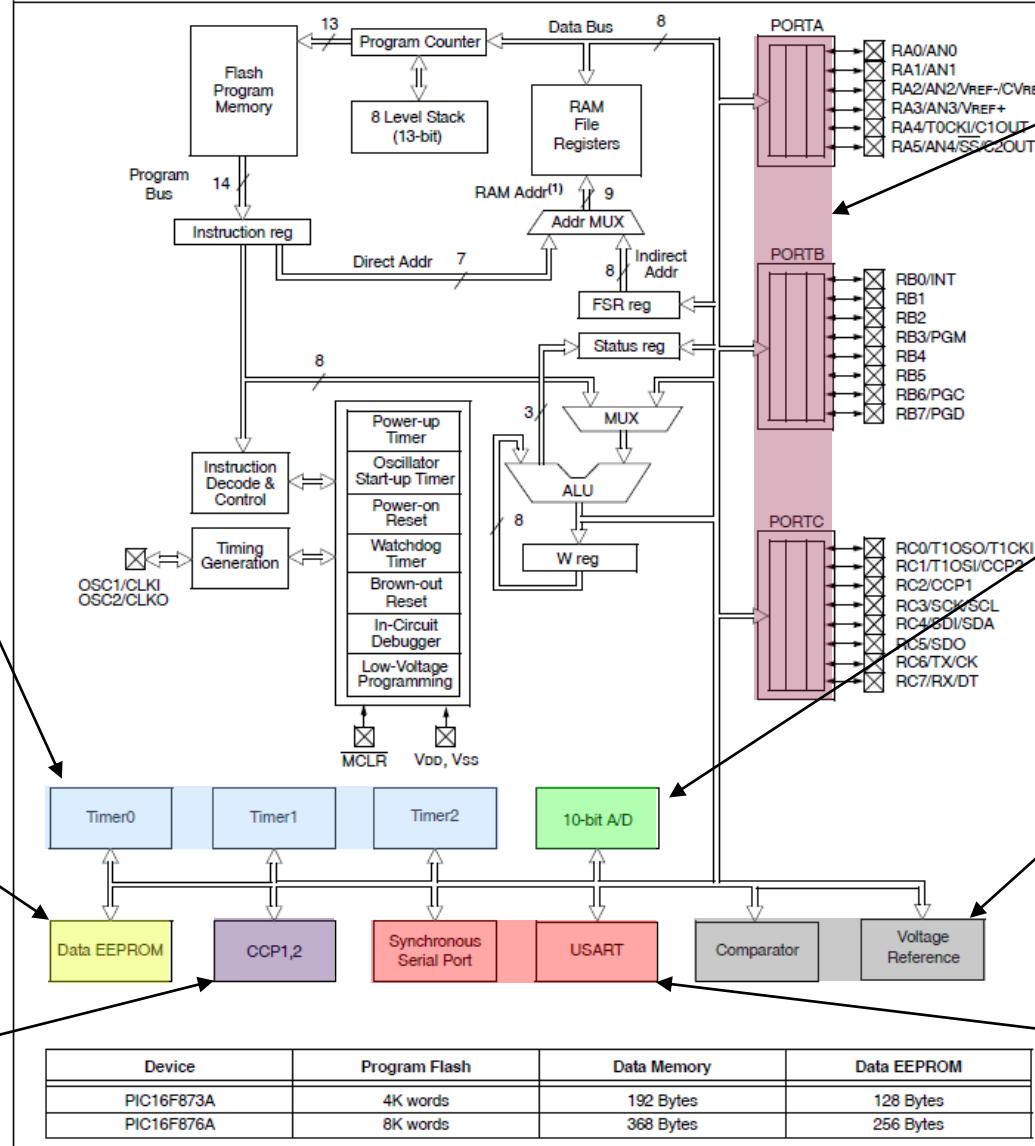
TABLE 1-1: PIC16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz			
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

1. Les micro-processeurs (et notamment les PIC)

Structure interne d'un PIC :

FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM



Fonction TIMER/COMPTEUR
Exécution de portion de code (macro) à un rythme paramétrable, par comptage d'événements externes ou internes (fronts d'horloge)

Fonction Mémoire EEPROM
Mémoire en lecture/écriture reprogrammable et non-volatile (durée de rétention des infos = 40 ans, 1 millions de cycle de lecture/écriture)

Fonction Capture/Compare et PWM
Détection d'un nombre d'événement extérieur atteint et possibilité de faire fonctionner une/des sorties logiques en MLI (Modulation de largeur d'impulsions)

Fonction Ports d'E/S

Permet au processeur d'échanger des informations **analogiques, logiques ou numériques** avec son environnement. Les ports sont paramétrables en entrées ou sortie.

Fonction CAN

Convertisseur Numérique/Analogique de résolution 10 bits, d'une ou plusieurs entrées analogiques, le résultat de la conversion sera placé dans une variable.

Fonction Comparateur et Tension de référence

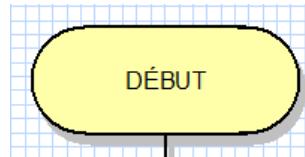
Comparateur de tensions présentes sur des entrées analogiques, le résultat de la comparaison pouvant être fourni sur une sortie logique. La comparaison peut-être réalisée avec une tension de référence interne.

Fonction Communication Série (UART, SPI et I²C)

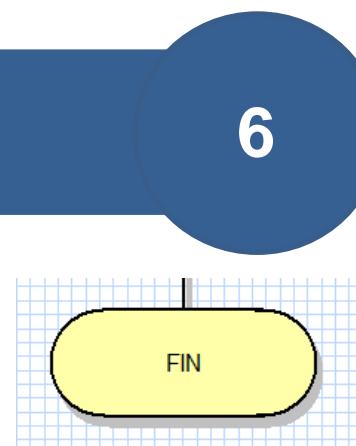
Confère à certaines E/S des fonctions de communication suivant différents protocoles série

Cl2 – SED, partie 2

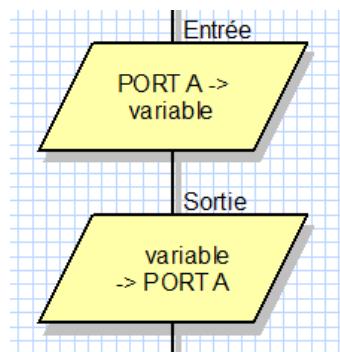
2. Les instructions de base (principaux blocs)



Chaque programme (principal ou « Main ») doit avoir un début et une fin.



Les sous-programme, ou « Macro » (ou procédures) respectent aussi cette règle.

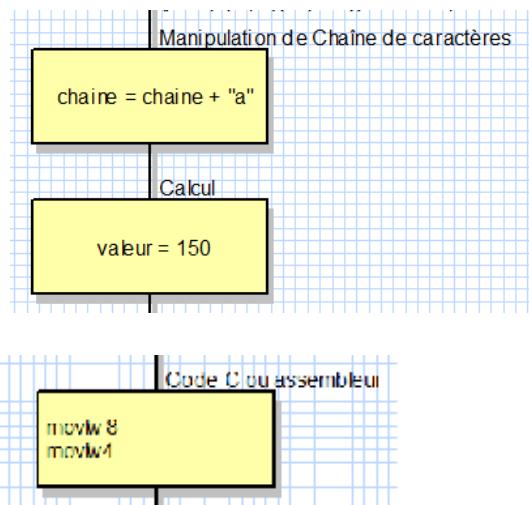


Les µC possèdent des ports d'E/S pour commander des « pré-actionneurs », ou connaître l'état de « capteurs ».

- Un bloc « Entrée » permet de relever l'état d'un bit d'un port, ou des 8 bits du port.

La valeur sera stockée dans une « variable » définie préalablement

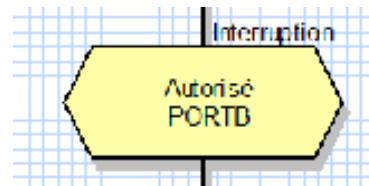
- Un bloc « Sortie » permet de forcer l'état d'un bit d'un port, ou l'état du port entier, d'après une valeur binaire, ou une valeur de variable.



Les rectangles permettent :

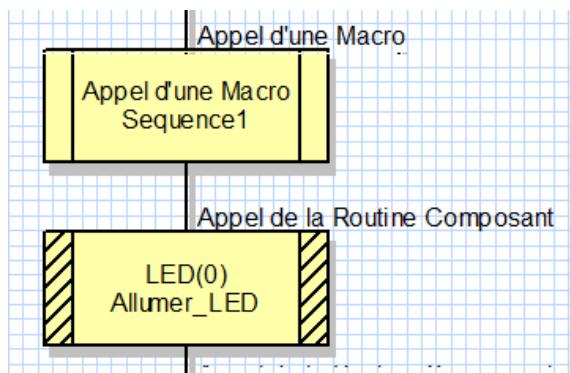
- de réaliser des manipulations de variables de type « chaîne de caractères » ou « STRING »
- de réaliser des calculs numériques sur des variables des différents types (« octet » de 0 à 255, « entier » de -32768 à 32767, ou « virgule flottante »)
- d'intégrer dans le logigramme des portions de code C, ou assembleur

2. Les instructions de base (principaux blocs)



Le blocs « Interruptions » permettent d'appeler une macro, lorsqu'un événement particulier se produit :

- Changement d'état d'une entrée d'interruption (INT)
- Changement d'état d'un bit d'un port d'E/S (PORTB par ex),
- Fin d'un cycle de comptage (TIMER),
- Arrivée de données sur port de communication série (I²C)

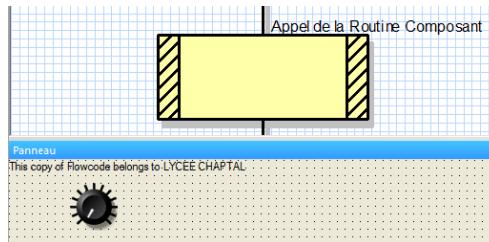


On doit respecter quelques règles d'organisation :

- le prog. « Principale », ou « Main » présente généralement une séquence d'appel à des « macro »,
- le programme principal n'est pas trop long,
- on réalise un ensemble d'actions ou de calcul au sein d'une « macro »,
- une bibliothèque de « macros » associée aux différentes cartes que l'on peut brancher sur les ports d'E/S, est disponible.

Cl2 – SED, partie 2

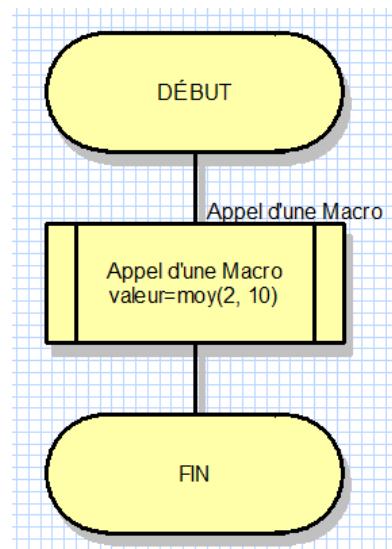
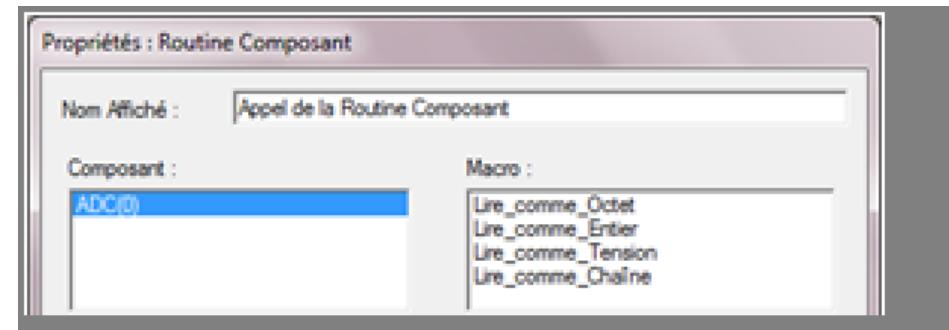
3. Appel de fonctions ou macros



On dispose d'un ensemble de fonctions, pour récupérer une image numérique d'une tension sur un port (ici ADC(0) pour le port RA0)

Exemple de « macros » ou « routine composant » :

Si on place un ADC (ou CAN en français) dans le panneau (de simulation en bas de l'écran).



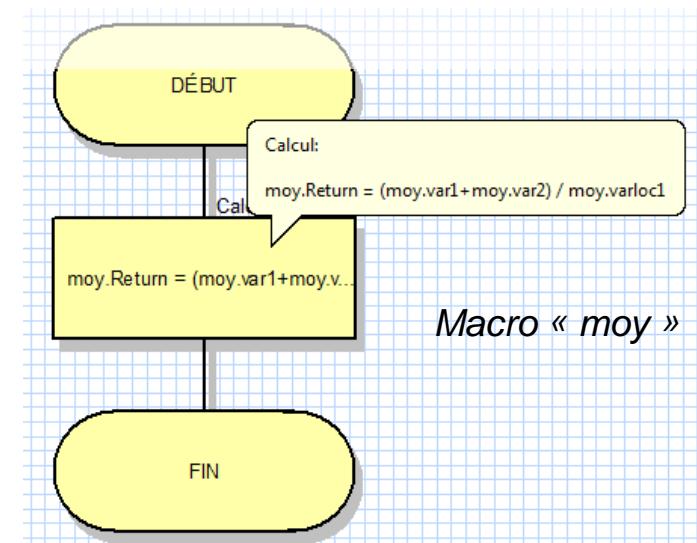
Prog. principal

Créer des « macros » avec des paramètres (ou des fonctions avec objets)

Lorsque la macro « moy » est appelée, les valeurs **2** et **10** sont respectivement associées aux « paramètres d'entrées » **var1** et **var2**.

La macro retourne le résultat du calcul (ou « paramètres de sortie » dans la variable **valeur**).

Le calcul est réalisée grâce à une variable locale **varloc1=2** (cette variable n'existe que pour la macro « moy »).



CI2 – SED, partie 2

4. Les structures décisionnelles

Les losanges permettent de réaliser des choix simples, ou multiples. Les conditions de passage vers une voie, plutôt qu'une autre sont de type booléen, (par exemple le résultat d'une comparaison d'une variable avec une valeur).

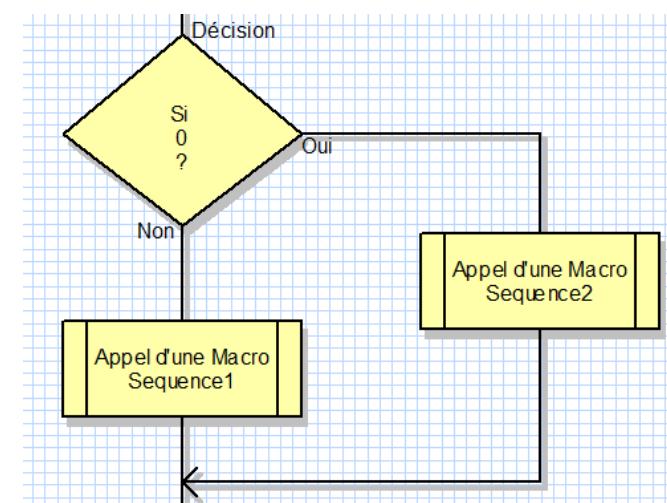
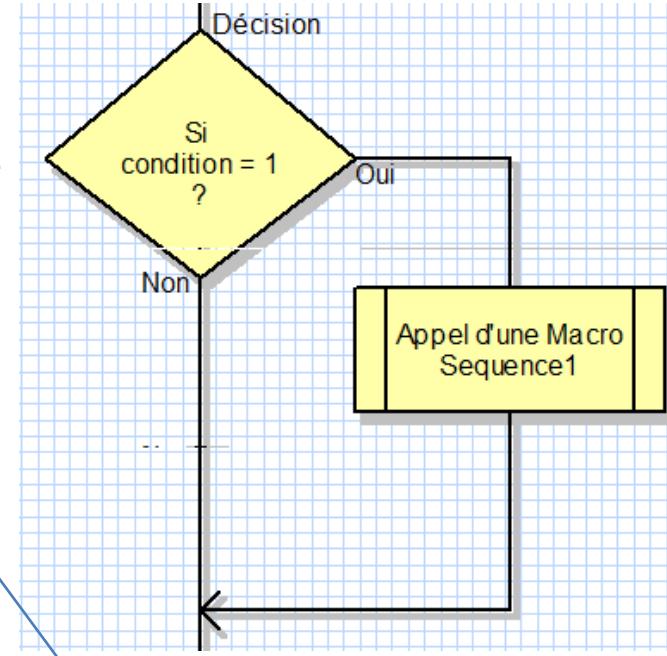
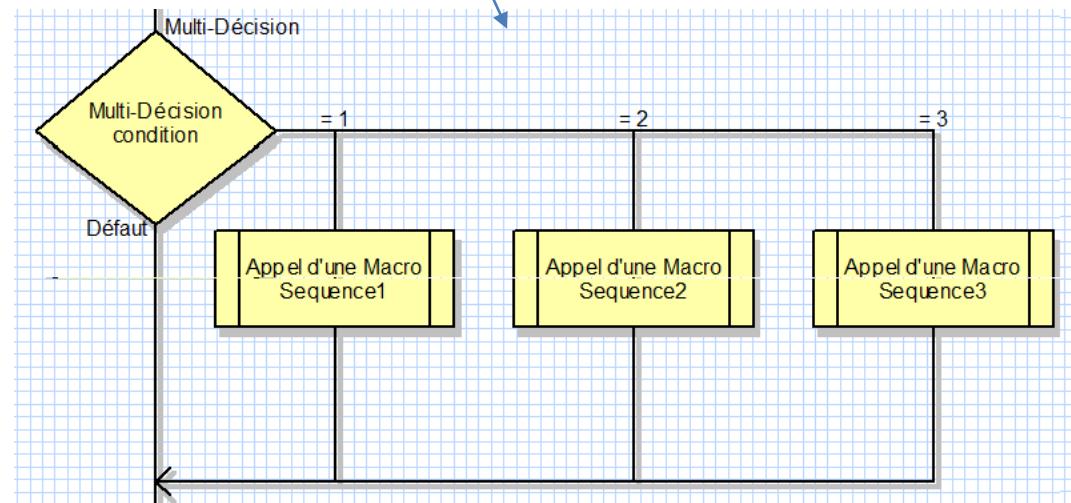
On distinguera 3 types de structure :

- **Si condition=vraie Alors action**
- **Si condition=vraie Alors action1 Sinon action2**
- **Cas**

Où condition1=vraie Alors action1

Où condition2=vraie Alors action2

...



5. Les structures de boucle

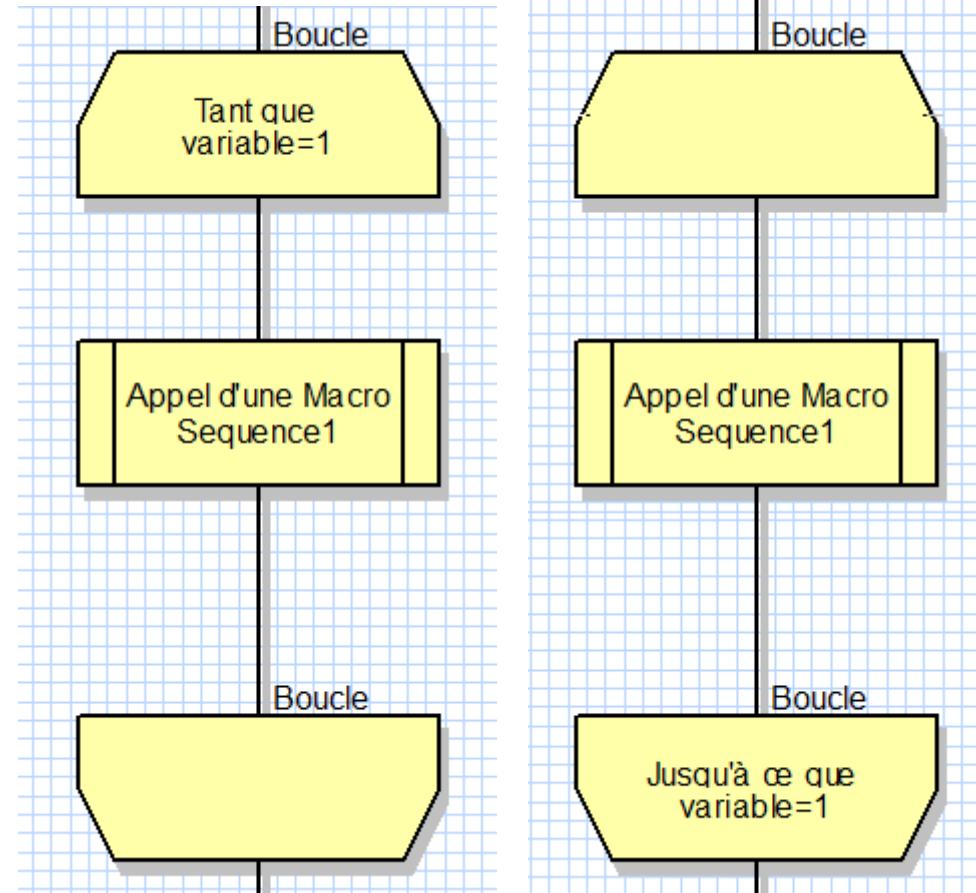
Les parallélépipèdes permettent de réaliser des boucles. Les conditions de sortie de la boucle sont aussi de type booléen.

On distinguera 2 types de structure :

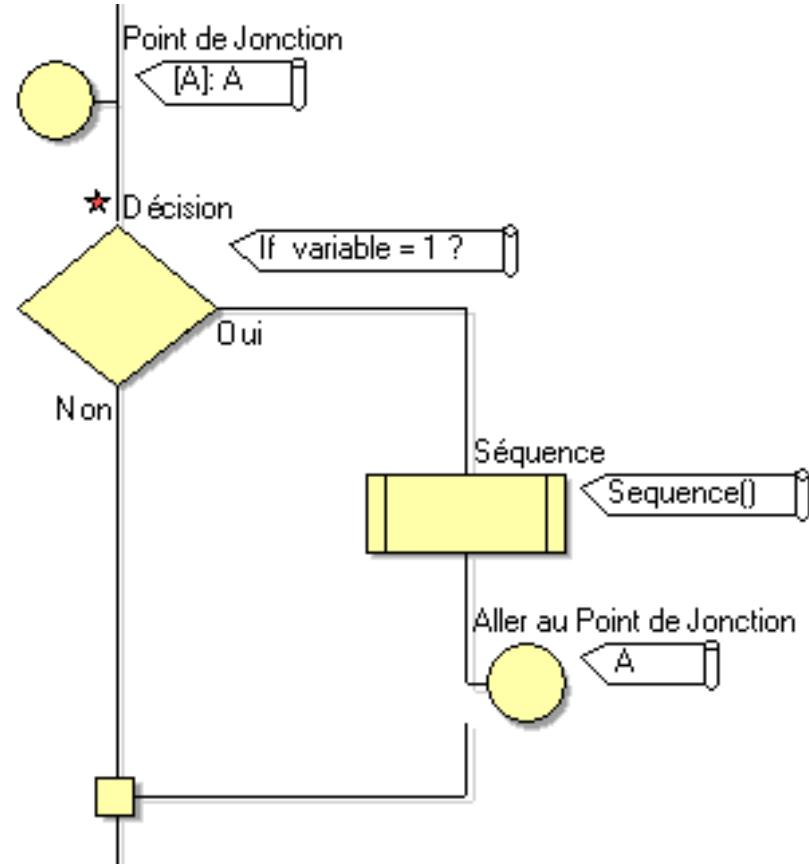
- **Tant que** condition=vraie **Alors** action
- **Répéter** action **Jusqu'à** condition=vraie

Remarque :

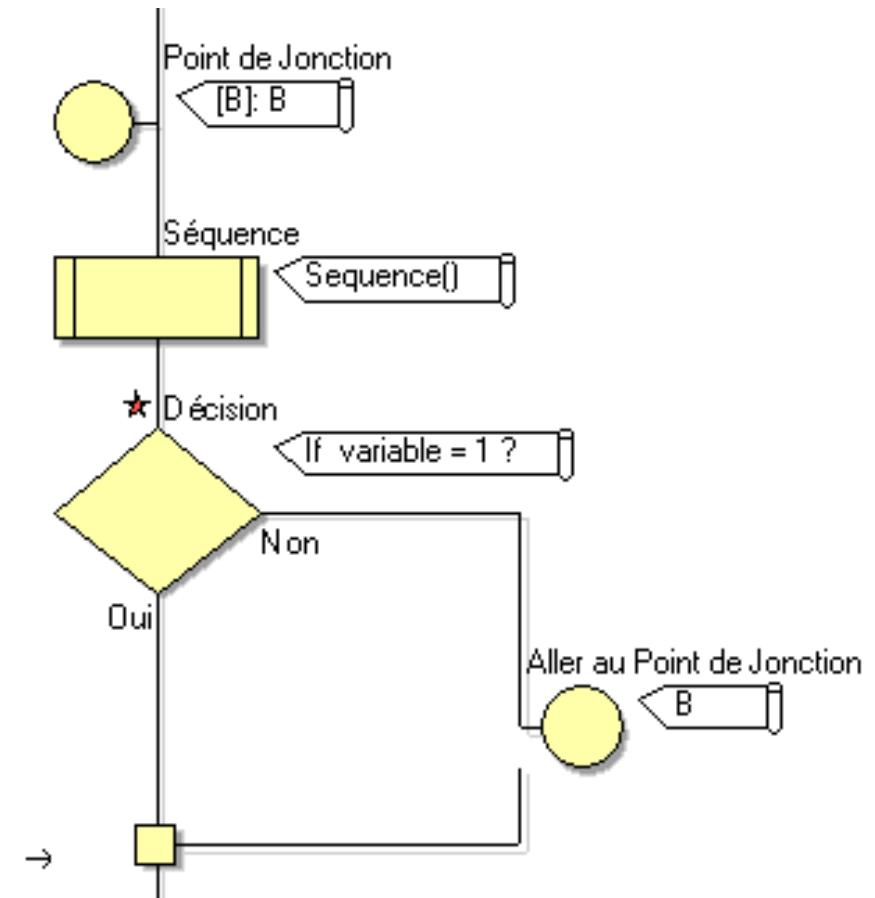
- On utilisera souvent des index qui permettront de compter le nombre de passage dans la boucle (variable qu'on incrémentera ou décrementera).
- On peut noter que suivant le cas, l'action est réalisée avant ou après le test.



Représentation plus classique avec des points de jonction :



Tant que condition=vraie
Alors action



Répéter action
Jusqu'à condition=vraie

6. Les fonctions de port d'E/S

L'ensemble des possibilités de ces blocs fonctionnels ne peut pas être abordé, car trop complexe. On prendra par exemple le bloc I/O PORTB du PIC16F876A. Des registres sont associés à ce port. Le contenu de ces registres est accessible par le microprocesseur en lecture et/ou en écriture suivant le cas. On peut aussi noter que chaque registre possédera une adresse distinct dans le « plan d'adressage » du microcontrôleur.

TABLE 4-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit 0	TTL/ST ⁽¹⁾	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit 1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit 2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM ⁽³⁾	bit 3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit 4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit 5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/PGC	bit 6	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or in-circuit debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit 7	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or in-circuit debugger pin. Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode or in-circuit debugger.

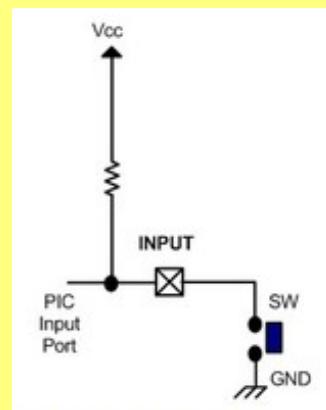
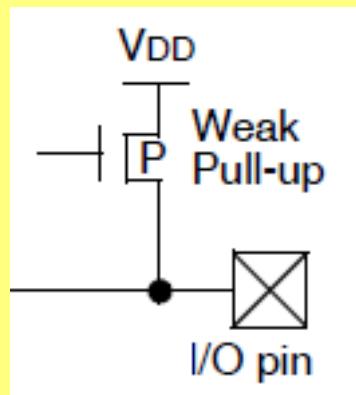
3: Low-Voltage ICSP Programming (LVP) is enabled by default which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 28-pin and 40-pin mid-range devices.

Les fonctions possibles du PORTB du PIC16F876A

6. Les fonctions de port d'E/S

Commentaires

- Le RB0 du PORTB peut-être utilisé pour lancer une macro d'«interruption». L'entrée (buffer) se voit alors conférée une fonctionnalité « Trigger de Schmitt » pour fixer un seuil de déclenchement.
- Les entrées/sorties peuvent être tirées au niveau 1 par un transistor équivalent (à l'état passant) à une forte résistance de « Pull-Up »



- Les bits RB3, RB6 et RB7 seront utilisés pour programmer le circuit (signaux nommés alors PGM, PGC et PGD)

Le microcontrôleur accède aux registres « TRISB » et « OPTION_REG » pour configurer le fonctionnement du PORTB. Il accédera au registre « PORTB » pour lire et/ou écrire une valeur sur les bits d'E/S

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
86h, 186h	TRISB	PORTB Data Direction Register							
81h, 181h	OPTION_REG	RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

6. Les fonctions de port d'E/S

Les registres associés au PORTB

TRISB

1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0
-----	-----	-----	-----	-----	-----	-----	-----

Le registre « TRISB » est le registre de direction :

- Un **bit n** du registre « TRISB » à **1** correspond à configurer la **broche n** du PORTB **en entrée**.
- Un **bit n** du registre « TRISB » à **0** correspond à configurer la **broche n** du PORTB **en sortie**

PORTB

PB7	PB6	PB5	PB4	PB3	PB2	PB2	PB1
-----	-----	-----	-----	-----	-----	-----	-----

Le port est un **port bidirectionnel**. On peut choisir si un bit est en « entrée » ou en « sortie ».

OPTION REG

RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
------	--------	------	------	-----	-----	-----	-----

Le **bit RBPU à 0**, provoque l'activation de toute les résistance de pull-up du PORTB. Cette configuration est possible lorsque le PORTB est configuré en entrée.

L'état du **bit INTEDG** va définir l'usage ou non de l'entrée RB0 comme entrée d'interruption externe