

ALGORITMIE

I) Présentation



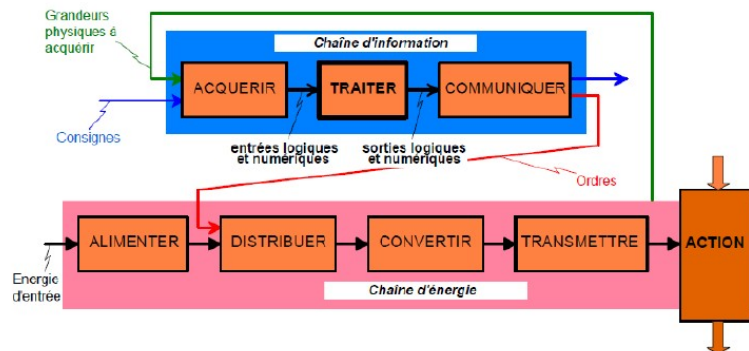
Ex : Microcontrôleur PIC16F877a

Un microcontrôleur (μC) est un composant programmable qui intègre dans un même boîtier :

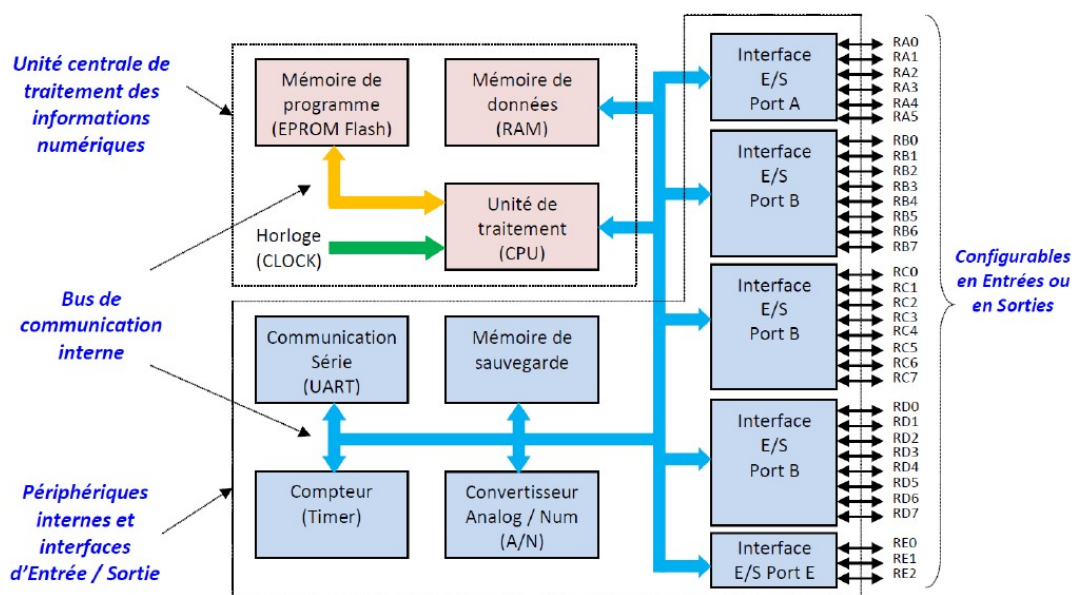
- un **microprocesseur (CPU ou μP)** qui traite des données logiques et arithmétiques et réalise le séquençage des opérations
- une **mémoire de programme**
- une **mémoire de données**
- des **interfaces d'entrée – sortie** qui permettent de communiquer avec des périphériques externes.

II) Identification de la fonction réalisée

Les systèmes numériques organisés autour de microcontrôleurs (PIC 16F877a, 68HC11...) réalisent la fonction **TRAITER** de la chaîne d'information :



III) Structure matérielle interne et simplifiée du PIC16F877a



3.1 L'unité de traitement (CPU : Central Processing Unit)

L'unité de traitement numérique exécute les instructions du programme (codées sur 14 bits). Il traite et produit des opérations sur des informations numériques uniquement.

3.2 Les mémoires

Elles se divisent en trois blocs distincts :

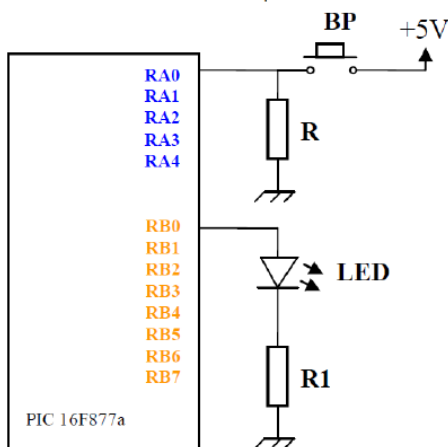
- la **mémoire de programme de type EPROM Flash** de capacité (8 kilo mots de 14 bits). Cette mémoire est destinée à être lue uniquement. Elle contient les ordres (instructions) que doit effectuer le CPU.
- la **mémoire de données de type RAM** de capacité (368 octets) est utilisée pour le stockage temporaire des données et résultats. Le CPU peut lire cette mémoire mais aussi modifier son contenu.
- la **mémoire de sauvegarde de type EEPROM** (256 octets) qui peut être lue et écrite depuis le programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.

3.3 L'horloge

Signal qui cadence l'exécution des instructions. Chacune des instructions du programme est traitée en un cycle machine : une période de l'horloge avec une division interne par 4. Exemple : avec une horloge externe de 8MHz, **la fréquence cycle est de 2 MHz soit une durée d'exécution d'une instruction de 500ns.**

3.4 Les ports d'entrée / sortie

Pour communiquer avec l'extérieur le PIC dispose de 5 ports (PORT A, PORT B, PORT C, PORT D et PORT E). Les ports sont bi-directionnels, ce qui signifie qu'ils peuvent être configurés et utilisés comme des entrées ou des sorties. Exemple :



L'état de la broche RA0 dépend de l'action sur BP

- Lorsque BP est ouvert **alors RA0 = 0**
Une lecture du registre PORTA donnera

	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
PORTA	-	-	-	x	x	x	x	0

- Lorsque BP est fermé **alors RA0 = 1**
Une lecture du registre PORTA donnera

	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
PORTA	-	-	-	x	x	x	x	1

L'état de la LED dépend de l'état logique présent sur la broche RB0.

Une écriture dans le registre PORTB donnera :

- Si RB0 = 0, **la LED sera éteinte**
- Si RB0 = 1, **la LED sera allumée**

	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
PORTB	x	x	x	x	x	x	x	0

	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
PORTB	x	x	x	x	x	x	x	1

Généralisation

Le microcontrôleur



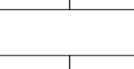
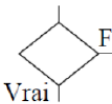

- ➔ Reçoit les informations sur un port d'entrée (Matériel)
- ➔ Traite ces informations (Logiciel)
- ➔ Utilise ces informations pour commander des circuits connectés sur un port de sortie (Matériel).

IV) Outils de description l'algorithme

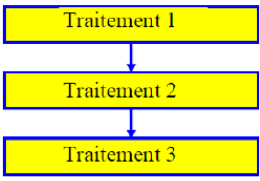
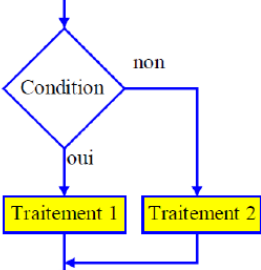
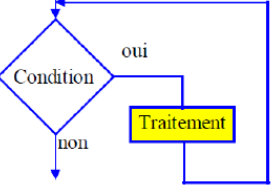
Un algorithme est un modèle universel de description d'un système numérique. Il existe deux types de représentation : « **l'organigramme** » et « **une représentation littérale algorithmique** ».

4.1 L'organigramme d'un programme

Un **organigramme** est la représentation graphique d'une suite structurée d'instructions. Voici les symboles graphiques normalisés à utiliser :

Symbole	Désignation	Symbole	Désignation
	Début, fin, interruption Début, fin ou interruption d'un programme		Lecture ou Ecriture d'une donnée externe Mise à disposition d'une information (écriture sur un port de sortie) ou enregistrement d'une information (lecture d'un port d'entrée).
	Traitement interne Opération ou calcul sur des données dont le résultat est stocké dans le microcontrôleur.	 Faux Vrai	Branchement Test, exploitation de conditions variables impliquant le choix d'une parmi deux. Symbole utilisé pour représenter une décision.
	Sous-programme Portion de programme considérée comme une simple opération.		

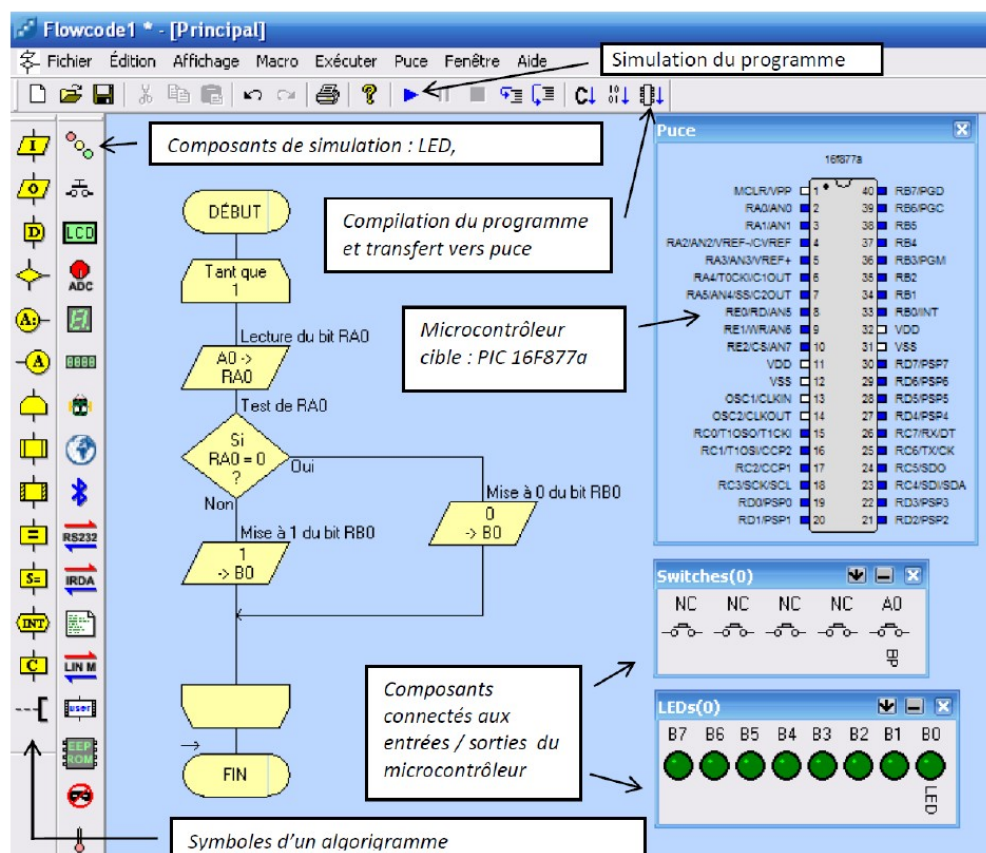
4.2 Les structures algorithmiques

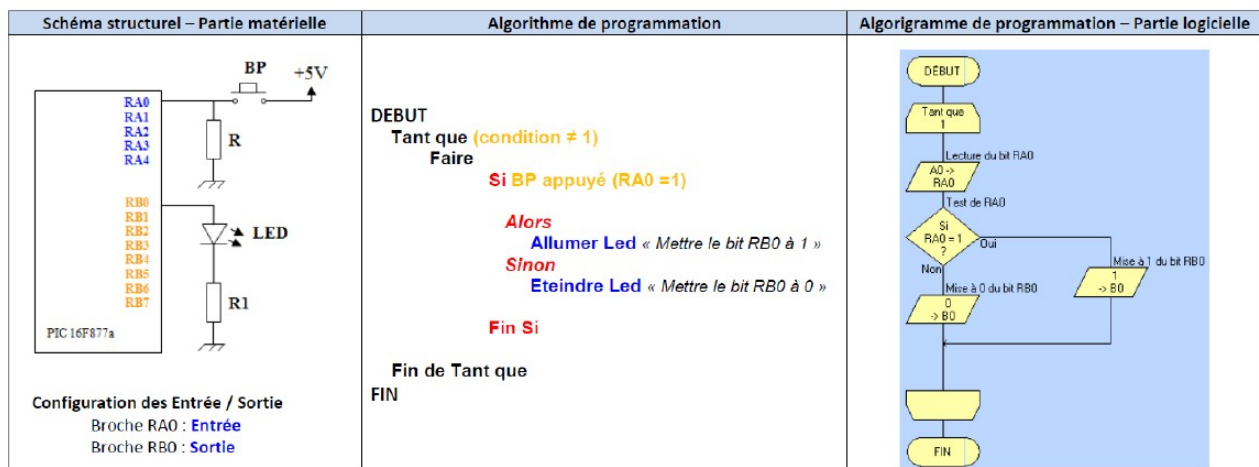
Structure	Organigramme	Algorithme
Séquence linéaire La structure linéaire se caractérise par une suite d'actions à exécuter successivement dans l'ordre de leur énoncé.		FAIRE « traitement 1 » FAIRE « traitement 2 » FAIRE « traitement 3 »
Séquence alternative ou conditionnelle Une structure alternative n'offre que deux issues possibles s'excluant mutuellement. Les structures alternatives définissent une fonction de choix ou de sélection entre l'exécution de l'un ou de l'autre des deux traitements.		SI « condition » vraie ALORS FAIRE « traitement 1 » SINON FAIRE « traitement 2 » FIN SI
Séquence répétitive ou itérative (boucle avec pré-test) Dans cette structure on commence par tester la condition, si elle est vraie alors le traitement est exécuté.		TANT QUE « condition » vraie FAIRE « traitement » FIN TANT QUE

Structure	Organigramme	Algorithme
Séquence répétitive ou itérative (boucle avec post-test) Dans cette structure le traitement est exécuté une première fois puis sa répétition se poursuit jusqu'à ce que la condition soit vérifiée.		REPETER « traitement » JUSQU'A « condition » vraie
Boucle avec comptage		POUR N = x A 0 REPETER « traitement » FIN POUR

V) Environnement de développement « FLOWCODE »

L'environnement de développement FLOWCODE utilise un langage graphique de haut niveau pour la programmation des microcontrôleurs PIC. Les programmes sont représentés sous forme d'algorithmes.



**Notion de variable**

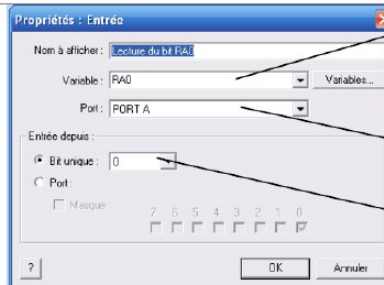
Une variable désigne une information qui peut changer au cours du programme. Sous FLOWCODE la taille d'une variable est de 8 bits : c'est donc un nombre qui pourra être compris entre 0 et 255. Utiliser une variable permet de :

- mémoriser une information provenant de l'extérieur,
- stocker le résultat d'une opération.

Notion de masque

Les masques permettent :

- de tester l'état d'un seul bit parmi les 8 bits d'un port, sans se préoccuper de l'état des 7 autres (en les masquant).
- d'écrire une valeur sur un seul bit parmi les 8 d'un port de sortie sans modifier les autres.



La variable RA0 est l'image de l'état logique appliqué à la broche A0 du Port A.

Lecture réalisée sur le Port A

Lecture unique du bit A0 et masquage des sept autres.