

1. Introduction

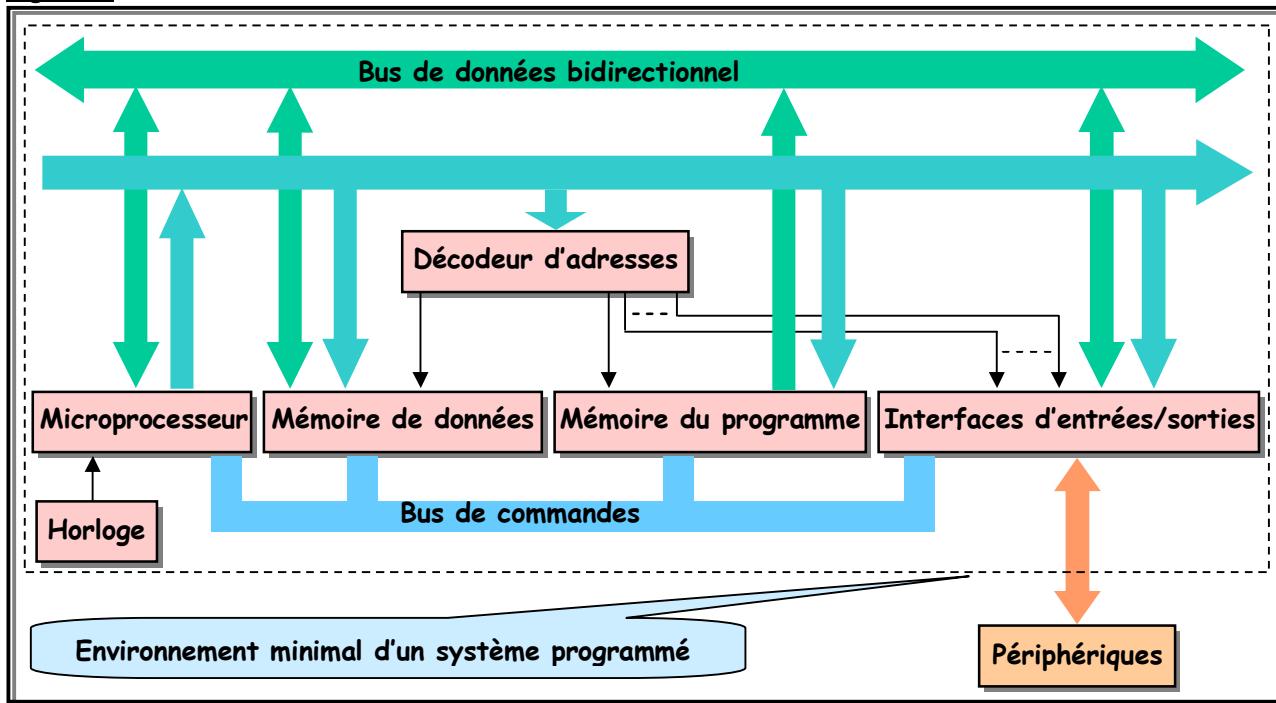
Le contrôle d'un processus industriel fait appel à un dispositif de gestion et de traitement des informations. Il est souvent réalisé par un système électronique programmé.

2. Architecture d'un système programmé

La structure matérielle d'un système programmé est essentiellement organisée autour d'un microprocesseur. Cependant, ce dernier ne peut fonctionner sans un environnement minimal.

Une telle structure est représentée à la figure 1.

Figure 1



21. Bus de données, d'adresses et de commandes

211. Notion de bus

Un bus est un ensemble de fils destiné à la communication entre les différents composants du système minimal. On distingue 3 groupes de bus : bus de données, bus d'adresses et bus de commandes.

212. Bus de données : Data bus

C'est un bus bidirectionnel qui assure le transfert des informations entre le microprocesseur et son environnement et inversement. Son nombre de lignes est égal à la capacité de traitement du μ P : 8, 16, 32 ou 64 bits.

213. Bus d'adresses : Adress bus

C'est un bus unidirectionnel qui permet la sélection du destinataire des informations à traiter dans un espace mémoire qui peut avoir 2^n emplacements où n est le nombre de conducteurs du bus d'adresses.

314. Bus de commandes : Control bus

C'est un bus constitué par quelques conducteurs qui assurent la synchronisation des flux d'informations sur le bus de données et d'adresse.

Exemple : Le μ P 6809 est doté d'un :

- Bus de données de 8 bits de D₀ à D₇.
- Bus d'adresses de 16 bits de A₀ à A₁₅.
- Bus de commandes de 10 bits : On cite à titre d'exemple les broches Read/Write, Halt et Reset.

22. Microprocesseur : CPU

C'est l'unité centrale de traitement (UCT ou CPU). Il exécute séquentiellement le programme (ensemble d'instructions) situé en mémoire ROM. L'exécution d'une instruction est cadencée par une horloge.

23. Mémoires

231. Mémoire du programme : mémoire morte

C'est une mémoire permanente et à lecture seule. Elle est destinée à stocker le programme. Il existe différents types, à savoir : ROM, PROM, EPROM ou UV EPROM, EEPROM et Flash EEPROM.

232. Mémoire de données : mémoire vive

C'est une mémoire volatile et à lecture/écriture. Elle est chargée de sauvegarder temporairement les données du programme (variables d'entrées, variables intermédiaires, variables de sorties, etc...). Il s'agit de la RAM.

24. Interfaces ou ports d'entrées/sorties

L'interfaçage est un ensemble matériel permettant d'assurer la liaison entre le μ P et les périphériques. On distingue les interfaces parallèles et les interfaces séries.

25. Périphériques

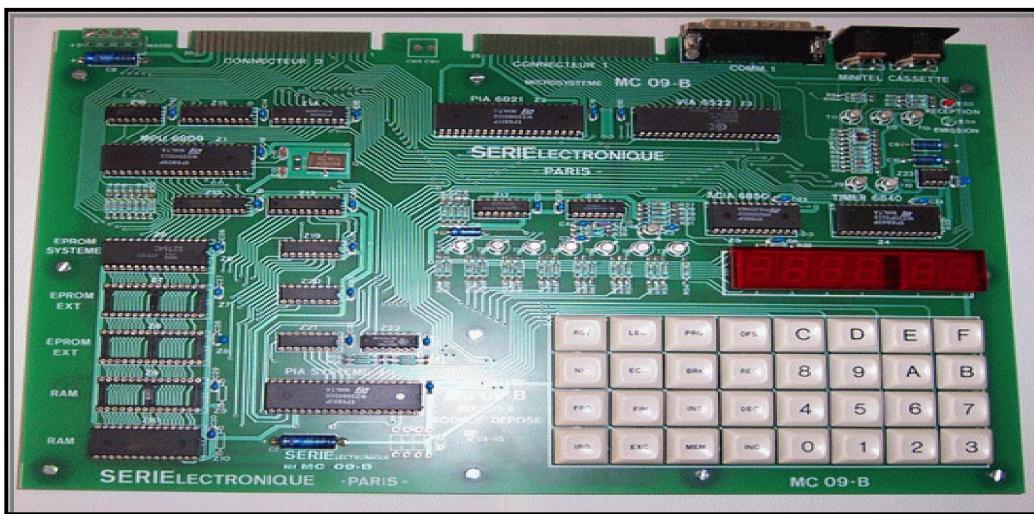
C'est un ensemble matériel extérieur au système programmé. On cite à titre d'exemple les interfaces homme/machine (clavier, afficheurs, interrupteurs, boutons poussoirs, etc...), capteurs et actionneurs.

26. Décodeur d'adresses

Le décodeur d'adresses est chargé d'aiguiller les données présentes sur le bus de données. En effet, le μ P peut communiquer avec les différents composants de son environnement. Ceux-ci sont tous reliés sur le même bus de données et afin d'éviter des conflits, un seul composant doit être sélectionné à la fois. Ainsi, à une adresse présentée par le μ P, devra répondre un seul composant du système (RAM, ROM, Interface).

Remarque : Lorsqu'un composant n'est pas sélectionné, il est mis en état haute impédance.

3. Kit didactique d'un système programmé à base du microprocesseur 6809



1. Introduction

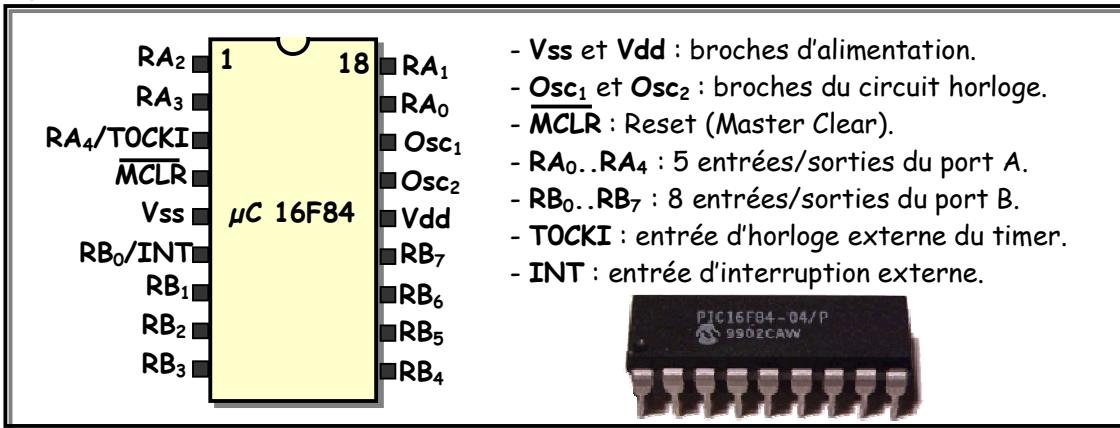
Le **PIC** (Programmable Interface Controller) **16F84** est un microcontrôleur produit par la société **MicroChip**. C'est un composant qui regroupe dans un même boîtier tous les éléments vitaux d'un système programmé : CPU, RAM, ROM, Interfaces d'entrées/sorties, etc.

2. Architecture du PIC 16F84

2.1. Organisation externe

Le μ C 16F84 est un circuit intégré de 18 broches : voir figure 1.

Figure 1

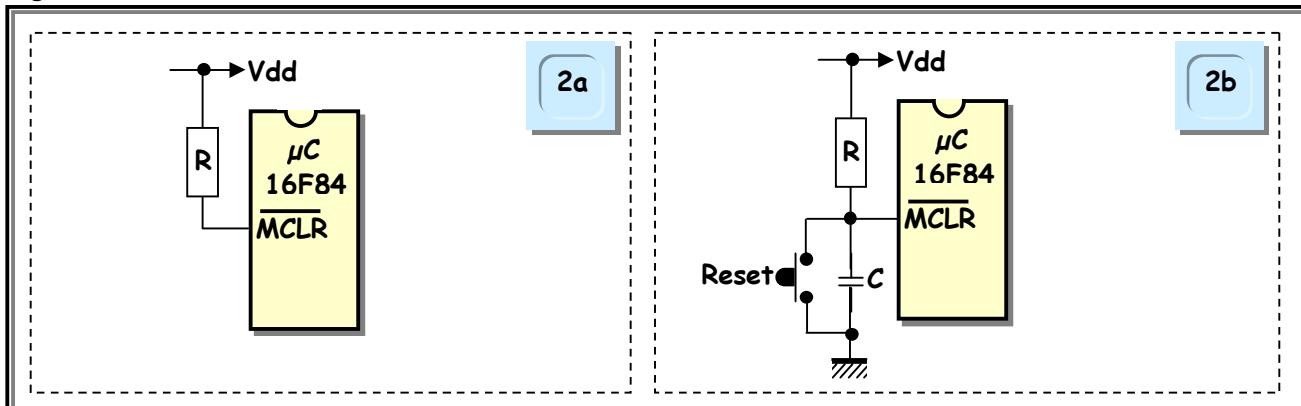


La tension nominale de l'alimentation du μ C 16F84 est +5 V : Vdd=+5 V et Vss=0 V.

Le **Reset** du μ C 16F84 peut avoir plusieurs causes :

- Une mise sous tension POR (Power On Reset) : voir figure 2a
- Une mise à 0 de la broche **MCLR** (Reset manuelle) : voir figure 2b.
- Un débordement du timer du chien de garde **WDT** : voir plus tard.

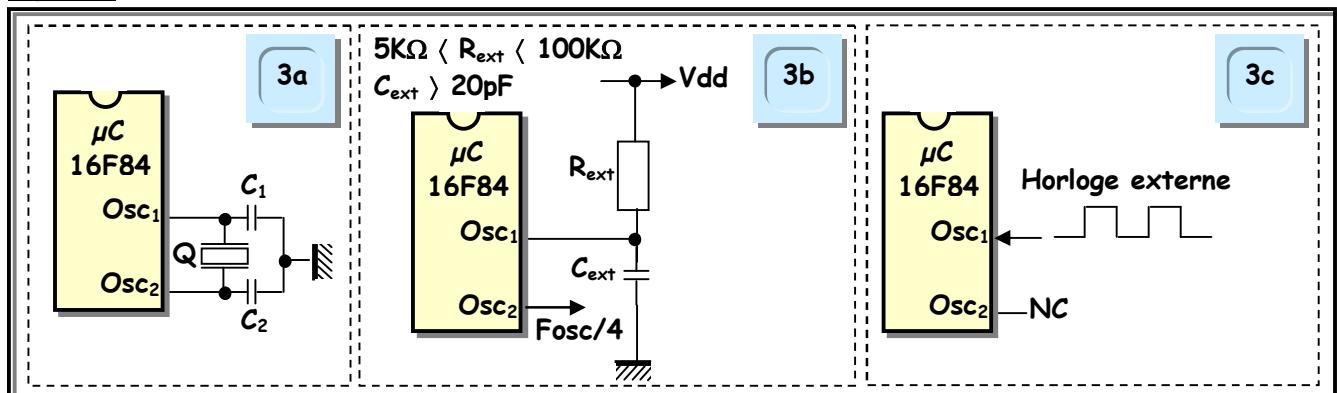
Figure 2



Le fonctionnement du μ C 16F84 nécessite une horloge qui rythme l'exécution des instructions du programme. On distingue trois modes d'horloge :

- Horloge interne à quartz** (Figure 3a) : Avec l'oscillateur à quartz, on peut avoir des fréquences allant de 4 MHz jusqu'à 20 MHz selon le type du μ C.
- Horloge interne à circuit RC** (Figure 3b) : Avec un oscillateur à circuit RC, la fréquence de l'oscillation dépend de la tension Vdd et des éléments R_{ext} et C_{ext} .
- Horloge externe** (Figure 3c) : Application d'un signal horloge externe.

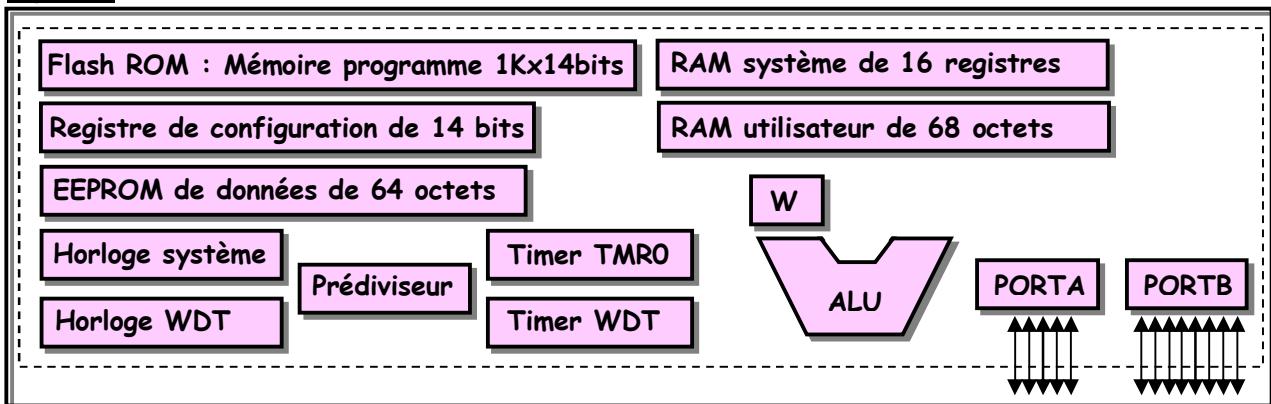
Figure 3



22. Organisation interne

La structure interne simplifiée du μ C 16F84 peut être donnée par la figure 4.

Figure 4



221. Horloge système

L'horloge système ou horloge instruction, est la base du temps interne qui cadence le fonctionnement du μ C. Quelque soit l'oscillateur utilisé, l'horloge système s'obtient en divisant la fréquence de l'oscillateur par **4**.

Exemple : Avec un quartz de 4 MHz, on obtient une horloge instruction de 1 MHz, soit le temps pour exécuter une instruction de 1 μ s.

222. Mémoire Flash ROM

Cette mémoire de **1024** mots de **14** bits stocke le programme. Elle est permanente et reprogrammable à souhait. Chaque position contient une instruction. Suite à un Reset ou lors d'une mise sous tension, le μ C commence l'exécution du programme à l'adresse **0**. De plus, lorsqu'il y a une interruption, il va à l'adresse **4**.

223. Registre de configuration

Pendant la phase de la programmation du μ C, on programme aussi un registre de configuration logé dans la mémoire EEPROM. Ce registre est un mot de **14** bits qui permet de :

- Choisir le type de l'oscillateur pour l'horloge.
- Valider ou non le timer du watchdog WDT.
- Autoriser ou non une temporisation à la mise sous tension.
- Interdire ou non la lecture des mémoires de programme et de données.

Registre de configuration \Rightarrow

CP	PWRTE	WDTE	FOSC ₁	FOSC ₀							
----	----	----	----	----	----	----	----	-------	------	-------------------	-------------------

- Bits **FOSC₀** et **FOSC₁** : Sélection du type d'oscillateur pour l'horloge.
FOSC₁FOSC₀=11 \Rightarrow Oscillateur à circuit RC jusqu'à 4 MHz.
FOSC₁FOSC₀=10 \Rightarrow Oscillateur HS, quartz haute fréquence, jusqu'à 20 MHz.
FOSC₁FOSC₀=01 \Rightarrow Oscillateur XT, quartz standard jusqu'à 4 MHz.
FOSC₁FOSC₀=00 \Rightarrow Oscillateur LP, quartz basse fréquence, jusqu'à 200 KHz.
- Bit **WDTE** : Validation du timer du watchdog WDT.
WDTE=1 \Rightarrow WDT validé et WDTE=0 \Rightarrow WDT inhibé.
- Bit **PWRTE** : Validation d'une temporisation à la mise sous tension.
Le μ C possède un timer permettant de retarder de 72 ms le lancement du programme après la mise sous tension. Ce délai maintient le μ C à l'arrêt et permet ainsi à la tension d'alimentation de bien se stabiliser.
PWRTE=1 \Rightarrow le μ C démarre tout de suite et PWRTE=0 \Rightarrow le μ C attend 72 ms.
- Bits **CP** : Protection en lecture des mémoires de programme et de données.
CP=1 \Rightarrow pas de protection et CP=0 \Rightarrow protection activée.

Exemple : On désire configurer le registre pour répondre aux critères suivants : oscillateur à quartz de 4 MHz, le timer du watchdog n'est pas autorisé, une attente de 72 ms est souhaitée et le μ C n'est pas protégé en lecture.

Configuration requise \Rightarrow

1	1	1	1	1	1	1	1	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---

 \Leftrightarrow 3FF1 en hexadécimal

224. Mémoire EEPROM de données

Cette mémoire est constituée de **64** octets que l'on peut lire et écrire depuis un programme. Ces octets sont conservés même après une coupure de l'alimentation et sont très utiles pour conserver des paramètres semi permanents : code d'accès, version du programme, message d'accueil, valeur invariable, etc.

225. Mémoire RAM

La mémoire RAM est constituée de deux parties : voir figure 5.

- Une zone mémoire de **68** octets réservée à l'utilisateur pour stocker ses variables. Elle adressable de l'@ 0x0C à l'@ 0x4F ou de l'@ 0x8C 0xCF.
- Un fichier de **16** registres de contrôle répartis en **2** banques :
 - Une banque **0** adressable de l'@ 0x00 à l'@ 0x0B.
 - Une banque **1** adressable de l'@ 0x80 à l'@ 0x8B.

L'accès à un registre s'effectue en se plaçant dans la banque où il se trouve. Ceci est réalisé en positionnant le bit **RPO** du registre d'état **STATUS** :

Registre STATUS \Rightarrow

	RPO	TO	PD	Z	DC	C
--	-----	----	----	---	----	---

RPO=0 \Rightarrow banque 0 et RPO=1 \Rightarrow banque 1.

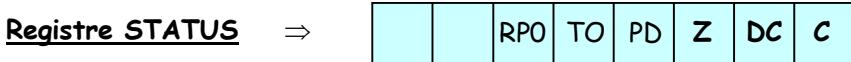
@	Banque 0	Banque 1	@
00	INDF	INDF	80
01	TMRO	OPTION	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07			87
08	EEDATA	EECON1	88
09	EEADR	EECON2	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C	Mémoire utilisateur	Idem	8C
4F			CF

226. ALU et le registre W

Le registre W, qui n'a pas d'adresse, est un registre de travail de **8 bits**.

L'ALU est une unité arithmétique et logique de **8 bits** qui réalise les opérations entre W et n'importe quel autre registre f ou constante k. Le résultat de l'opération peut être placé soit dans W soit dans f.

L'ALU est associée au registre d'état **STATUS** par les bits Z, C et DC :



- C** : Ce bit flag passe à 1 lorsqu'il y a une retenue sur un octet.
- DC** : Ce bit flag passe à 1 lorsqu'il y a une retenue sur un quartet.
- Z** : Ce bit flag passe à 1 quand le résultat d'une opération est nul.

227. Port d'entrées/sorties PORTA

Le port A désigné par **PORTA** est un port bidirectionnel de **5 bits** (RA_0 à RA_4). La configuration de direction pour chaque bit du port est déterminée avec le registre **TRISA** :

- Bit i de $TRISA=0 \Rightarrow$ bit i du PORTA est configuré en sortie.
- Bit i de $TRISA=1 \Rightarrow$ bit i du PORTA est configuré en entrée.

Les broches RA_0 à RA_3 sont des entrées/sorties compatibles TTL alors que la broche RA_4 peut être utilisée soit comme entrée/sortie normale du port A, soit comme entrée horloge externe pour le Timer TMRO. Le choix se fait à l'aide du bit **TOCS** du registre **OPTION** :

- $TOCS=0 \Rightarrow RA_4$ est une entrée/sortie normale.
- $TOCS=1 \Rightarrow RA_4$ est une entrée horloge externe pour le timer TMRO.

De plus, RA_4 est une sortie à drain ouvert. Donc, il ne faut pas oublier de mettre une résistance externe vers Vdd.

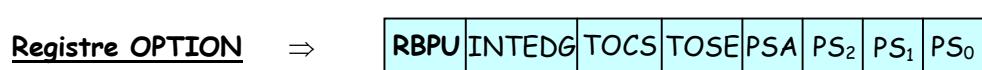
Chaque broche du port A configurée en sortie peut fournir un courant de 20 mA au maximum, mais tout le port A configuré en sortie ne peut pas débiter un courant total supérieur à 50 mA.

Chaque broche du port A configurée en entrée peut accepter un courant de 25 mA au maximum, mais tout le port A configuré en entrée ne peut pas accepter un courant total supérieur à 80 mA.

228. Port d'entrées/sorties PORTB

Le port B désigné par **PORTB** est un port bidirectionnel de **8 bits** (RB_0 à RB_7). La configuration de direction se fait à l'aide du registre **TRISB** (identique à celle du PORTA : voir TRISA). Toutes les broches sont compatibles TTL.

Les entrées du port B peuvent être pourvues de résistances de rappel à Vdd. Cette option est validée par le bit **RBPU** du registre **OPTION** :



$RBPU=0 \Rightarrow$ les broches configurées en entrées sont connectées à des résistances de rappel à Vdd et les broches configurées en sorties sont dépourvues automatiquement de ces résistances de rappel.

En entrée, la broche RB_0 appelée aussi INT peut déclencher l'interruption externe INT et une quelconque des broches RB_4 à RB_7 peut déclencher l'interruption RBI.

Chaque broche du port B configurée en sortie peut fournir un courant de 20 mA au maximum, mais tout le port B configuré en sortie ne peut pas débiter un courant total supérieur à 100 mA.

Chaque broche du port B configurée en entrée peut accepter un courant de 25 mA au maximum, mais tout le port B configuré en entrée ne peut pas accepter un courant total supérieur à 150 mA

 **Remarque** : Au Reset du μ C, les ports PORTA et PORTB sont configurés en entrée.