ECRITURE D'UN TEXTE SUR UN AFFICHEUR LCD

L'afficheur LCD (Liquid Crystal Display) possède 2 lignes de 16 caractères. Chaque caractère est représenté par une matrice de 5x8 points. Les caractères doivent être codés selon le code ASCII (American Standard Code for Information Interchange). Il comporte 16 broches :

8 broches reliées au bus de données (Db0 à Db7).

3 broches de commande (RS, RW, E).

2 broches d'alimentation (0, +5V).

3 broches pour le rétro-éclairage et le réglage du contraste.

Les liaisons avec le µC sont :

- *Le bus de données de 8 bits*

LCD	*Db0*	*Db1*	*Db2*	*Db3*	*Db4*	*Db5*	*Db6*	*Db7*
Couleur	*Jaune*	*Vert*	*Bleu*	*Jaune*	*Vert*	*Bleu*	*Jaune*	*Vert*
µC	*P2.0*	*P2.1*	*P2.2*	*P2.3*	*P2.4*	*P2.5*	*P2.6*	*P2.7*

- *L*
  Le bus de commande de 3 bits

LCD	*RS*	*RW*	*E*
Couleur	*Jaune*	*Vert*	*Bleu*
µC	*P1.5*	*P1.6*	*P1.7*

Initialisation

Pour fonctionner correctement, l'afficheur nécessite une initialisation , contenue dans le sous programme :

init_lcd :

qui :
- *permet d'afficher sur 2 lignes de caractères de taille 5x8 points*
- *définit le sens de progression de l'écriture vers la droite*
- *allume l'afficheur après une temporisation fournie par le Timer0.*

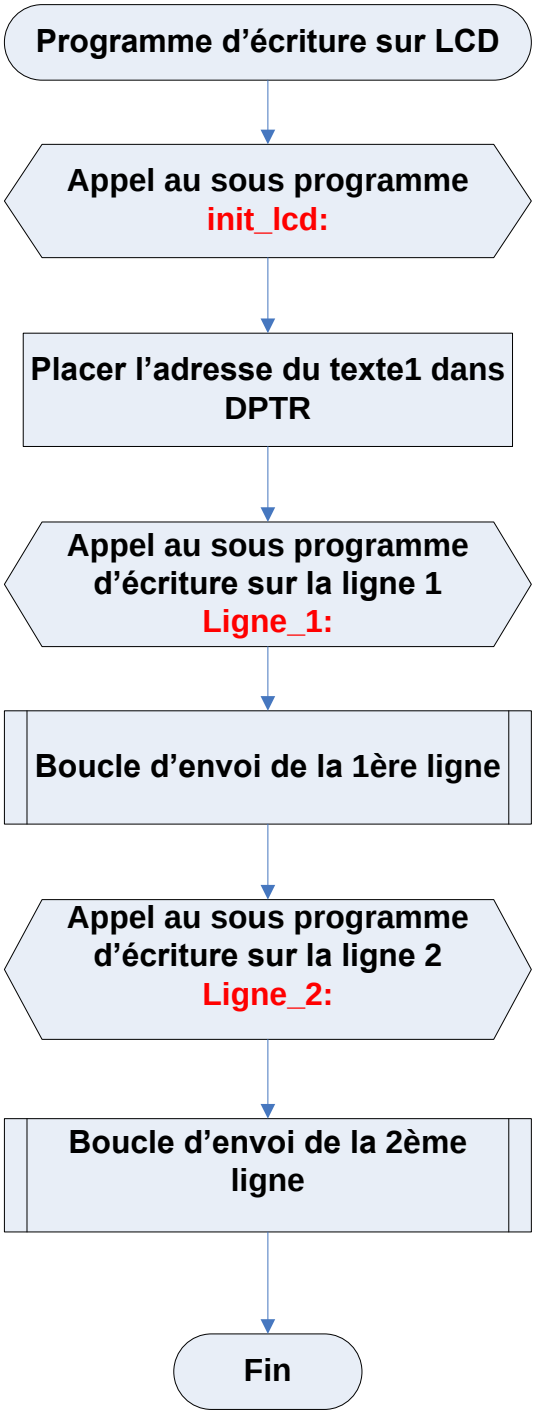
Envoi d'une donnée

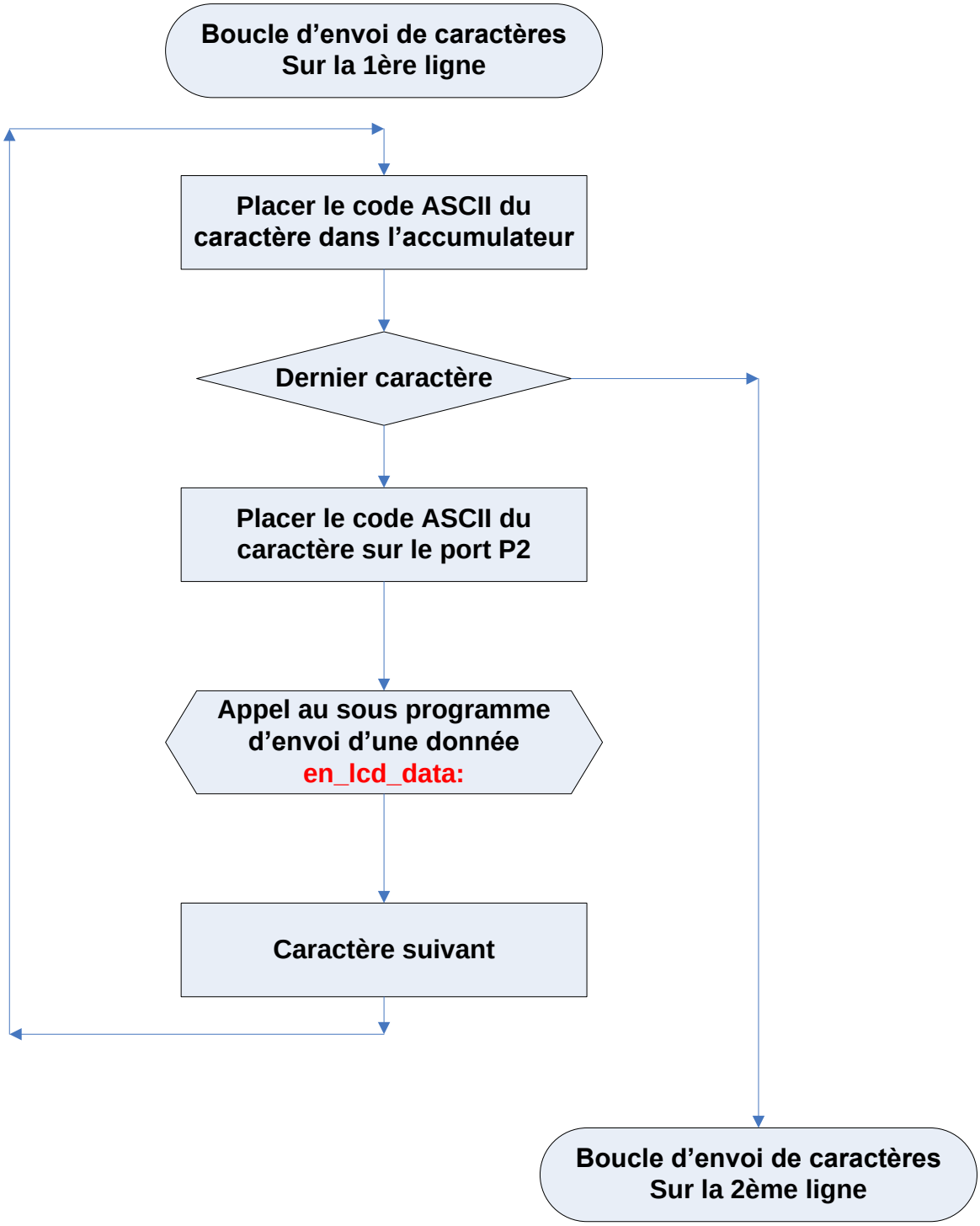
1- *Placer le code hexadécimal de la 'donnée' sur le port P2 relié au LCD*
2- *Appeler le sous programme d'envoi de commande : en_lcd_data :*

Structure du programme assembleur

```
;       titre et
;
;description du programme
;-------------------------------------------------
;Zone de définition des variables
;
;1 les variables de l'afficheur LCD
;-------------------------------------------------
;1' vos propres variables
;-------------------------------------------------
;2 Zone des sous programmes d'interruption
; de 0000h à 002Fh
        org     0000h
        ljmp    debut
        ;emplacement des autres programmes
        ;d'interruption
;-------------------------------------------------
;3 Les sous programmes du LCD
        org     0030h
  init_lcd:     ;initialisation du LCD
  tempo:        ;temporisation
  en_lcd_code:  ;envoi d'une commande
  test_busy_lcd: ;attente de la réponse du LCD
  en_lcd_data:  ;envoi d'une donnée
  ligne_1:      ; ecriture sur la ligne 1
  ligne_2:      ; ecriture sur la ligne 2

;-------------------------------------------------
;4 Zone des textes à envoyer
        org     00F0h
texte1:
        db      'TEXTE1'
        db
texte2:
        db      'TEXTE2'
        db
;-------------------------------------------------
;5 Vos sous programmes
        org     0180h
;-------------------------------------------------
;6 Programme principal
        org     0220h
debut:
        end
```


Programme d'écriture sur LCD
Appel au sous programme
init_lcd:
Placer l'adresse du texte1 dans
DPTR
Appel au sous programme
d'écriture sur la ligne 1
Ligne_1:
Boucle d'envoi de la 1ère ligne
Appel au sous programme
d'écriture sur la ligne 2
Ligne_2:
Boucle d'envoi de la 2ème
ligne
Fin



Simulation du programme d'écriture

Mettre en commentaire la ligne d'appel au sous programme *init_lcd*

Compiler

Activer la simulation (menu ***Debug****)*

Placer un point d'arrêt sur la ligne de code ***clr A*** *située avant l'envoi du 1er caractère au LCD*

Activer le mode animé

Exécuter la simulation avec la vitesse maximale [Speed réglée sur le maximum et touche GO (dans la barre d'icones)]

Le programme attend une réponse du LCD (Busy Flag)

Arrêter la simulation [STOP]

Placer « 0 » sur P2.7 pour simuler cette réponse

Relancer la simulation

Arrivé au point d'arrêt : Valider l'observation des chronogrammes en mode continu avec 400 points enregistrés (menu View → Trace)

Placer une sonde sur P2

Oter le point d'arrêt

Observer la succession des codes envoyés au LCD