Fichier source trame de cet additif à la séquence 6 : LCDManuelEleve.c

Vous venez de réaliser la séquence 6. Vous avez utilisé les fonctions Printf et gotoxy pour afficher des messages sur l'afficheur. Maintenant vous allez réaliser une commande « à la main » de cet afficheur.
L'objectif : afficher « Ca marche ! » sur l'écran du LCD.

La fonction Ecrire_Message()

Le message sera dans un tableau :
int8 MESSAGE[17]=" Ca marche ! "; //Remarque : Ajouter 1 aux total des caractères.

Positionner RS au niveau haut ou bas ?
Positionner EN au niveau bas
Faire 16 fois :
- Placer la valeur contenu dans le tableau pointé par ***indice*** dans une variable ***var***
- Déplacer le quartet MSB de ***var*** en position LSB
- Imposer RS=1 dans la variable ***var***
- Écriver cette variable sur les 4 bits du LCD
- Valider
- Placer la valeur contenu dans le tableau pointé par ***indice*** dans une variable ***var***
- Imposer EN=0 et RS=1 dans la variable ***var***
- Écriver cette variable sur les 4 bits du LCD
- Valider

Fin de Faire 16 fois

☝ Fonction Valider : Une impulsion sur EN=1 pendant **1us** suivi de En=0 pendant **50us**.

☝ Imposer **EN=0** et **RS=1** : lors de la validation des données présentent sur les 4 bits, (EN=1 pendant 1us), il est obligatoire que **EN=0** et **RS=1** (piste : fonction ***bit_clear(xx,xx)*** *et* ***bit_set(xx,xx)***)

- ➔ Codez en C cette fonction.
- ➔ Effectuez une simulation sous ISIS afin de valider votre fonction en mode pas à pas.
- ➔ Justifiez la raison pour laquelle nous n'impose pas **EN=0** avant l'écriture du MSB sur les 4 bits du LCD alors que dans la 2° partie de la fonction, c'est le cas.

➔ **La fonction Init_LCD()**

Début :
- Positionner RS au niveau haut ou bas ?
- Patienter **xx**ms min après la mise sous tension
- Écrire sur les 4 bits du LCD la valeur **0x3**
- Valider
- Écrire sur les 4 bits du LCD la valeur **0x3**
- Valider
- Écrire sur les 4 bits du LCD la valeur **0x2**
- Valider
- Écrire sur les 4 bits du LCD la valeur **0xyy**
- Imposer EN=0 et RS=0
- Valider
- Écrire sur les 4 bits du LCD la valeur **0xyy**
- Imposer EN=0 et RS=0
- Valider
- Écrire sur les 4 bits du LCD la valeur **0xyy**
- Imposer EN=0 et RS=0
- Valider
- Écrire sur les 4 bits du LCD la valeur **0xyy**
- Imposer EN=0 et RS=0
- Valider

Fin

☝ **YY** :

[Mode fonctionnement] : 4bits, 2 lignes, 5x7 pixels
[Mode d'affichage] : Afficheur en fonction - Curseur visible - Clignotement du caractère
[Mode d'entrée] : Incrémentation du curseur (adresse) - Écran fixe.
[Effacement Affichage]: Mémoire affichage effacé - curseur en position 00 (home).

☝ Fonction Valider = ***valid_LCDlong()*** : Une impulsion sur **EN=1** pendant **1us** suivi de **En=0** pendant **2ms**

☝ Il est évident que pour coder cette fonction, il est judicieux d'effectuer la lecture d'un tableau.

➔ En vous inspirant de la fonction ***Ecrire_Message()***, rédigez l'algorithme de la fonction en justifiant toutes les valeurs.
➔ Codez cette fonction en C. Vérifiez la à l'aide d'une simulation sous ISIS en mode pas à pas.
➔ Justifiez les valeurs **1us** et **2ms** de la fonction ***valid_LCDlong()***
➔ Validez votre programme sur votre carte de thème.
➔ Éclairer les leds de sorte que l'on visualise l'exécution dans la fonction ***Init_LCD()*** et ***Ecrire_Message()***
➔ Afficher votre nom sur la 2° ligne (piste : envoyer une commande 0xC0 avec (RS=0))