* Driver LCD

Message boot

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | W | R | - | P | O | W | E | R |  | m | e | t | e | r |  |
| F | 8 | K | G | L |  |  |  |  |  |  |  | v | 0 | . | 1 |

Tempo : 5s

Message de mesure

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F | W | D |  |  | R | E | F |  |  | S | W | R |  |  |  |
| 1 | 0 | 0 | W |  | 1 | 0 | 0 | W |  | 1 | . | 1 | 4 | ! | ! |

« !! » si le SWR > 2

Message de calibration

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | D | C | f | w | d |  | x | x | x | x |  |  |  |  |  |
| A | D | C | r | e | f |  | x | x | x | x |  |  |  |  |  |

* Main
  + Afficher le message de boot pendant 5s
  + Tester le mode calibration (strap de soudure à la masse ou VCC)
  + Si le boitier est en mode de calibration
    - Dans une boucle infinie, lire les registres ADCfwd et ADCref
    - afficher le message de calibration
  + Sinon, dans une boucle infinie :
    - lire les registres ADCfwd et ADCref
    - calculer la puissance FWD et REF
    - Calculer le SWR
    - Afficher le message de mesure
* Fonctions
  + Afficher une chaine de caractère sur le LCD
  + Lire les registres ADC
  + calculer la puissance fwd et la puissance ref

|  |  |
| --- | --- |
| Paramètres entrée | ADCfwd, ADCref  table de calibration stockée en mémoire |
| Paramètres sorties | P\_FWD  P\_REF |
| Traitements | Lire la valeurs dans les registres ADC  récupérer la valeur correspondante dans la table de calibration |

* + calculer le swr

|  |  |
| --- | --- |
| * Paramètres entrée | * P\_FWD * P\_REF |
| * Paramètres sorties | * SWR |
| * Traitements |  |

* Application de calibration

A l'aide 3 points de mesures (1W, 5, 50W), interpolation de la courbe de calibration

Définir la procédure de calibration :

-avec le « petit détecteur maison », faire le relevé de la courbe P = f(ADC) sur les 2 ports FWD et REF

-avec la ligne de couplage complète, relever 3 points de mesures ADC = f(P)

-faire une interpolation entre les points qui tend à s'approcher de la 1ère courbe

* /prj
  + Makefile
  + Main.asm

|  |  |
| --- | --- |
| Fonctions | Fonction principale, point d’entrée du logiciel |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements | * Initialisation   + PIC TBD   + LCD : Effectuer l’initilisation du LCD (lcd\_init)   + ADC : TBD * Afficher le message de boot (lcd\_affbootmsg) * Tempo de 5s (TBD) * Tester le mode calibration (TBD strap de soudure à la masse ou VCC) * Si le boitier est en mode de calibration (TBD)   + Dans une boucle infinie     - lire les registres ADCfwd et ADCref     - afficher le message de calibration (lcd\_affcalib TBD) * Sinon   + Dans une boucle infinie :     - lire les registres ADCfwd et ADCref     - calculer la puissance FWD et REF     - Calculer le SWR     - Afficher le message de mesure |

* /sw/lcd
  + lcd.asm :

|  |  |
| --- | --- |
| Fonctions | Initialisation du LCD |
| Nom | lcd\_init |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements |  |

|  |  |
| --- | --- |
| Fonctions | Affichage d'un caractère |
| Nom | lcd\_affchar |
| Paramètres entrée | W(1 byte) : contient le caractère à afficher à la position courante du curseur |
| Paramètres sorties |  |
| Traitements |  |

|  |  |
| --- | --- |
| Fonctions | Envoi d’une commande au LCD |
| Nom | lcd\_sendcmd |
| Paramètres entrée | W(1 byte) : contient la commande  0x28 Set Interface Length  0x10 Turn Off Display  0x01 Clear Display RAM  0x06 Set Cursor Movement  0x0C Turn on Display/Cursor  0x01 Clear display  0xc0 move to 2nd row, first column |
| Paramètres sorties |  |
| Traitements |  |

|  |  |
| --- | --- |
| Fonctions | Positionner le curseur du LCD |
| Nom | lcd\_setposcursor |
| Paramètres entrée | W(1 byte) : contient la position du curseur  0-15 : 1ère ligne  16-31 : 2ème ligne |
| Paramètres sorties |  |
| Traitements | 1. Si le curseur doit être positionné sur la première ligne :   W = W + 0x80  Si le curseur doit être positionné sur la deuxième ligne :  W = W + 0xC0   1. Envoi de la commande au LCD (lcd\_sendcmd) |

|  |  |
| --- | --- |
| Fonctions | Efface le LCD |
| Nom | lcd\_clear |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements | 1. W=0x01 2. Envoi de la commande au LCD (lcd\_sendcmd) |

|  |  |
| --- | --- |
| Fonctions | Positionne le curseur sur la 2ème ligne |
| Nom | lcd\_setposL2 |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements | 1. W=0xC0 2. Envoi de la commande au LCD (lcd\_sendcmd) |

|  |  |
| --- | --- |
| Fonctions | Conversion hexa-ASCII |
| Nom | lcd\_convtoascii |
| Paramètres entrée | W (1 quartet) : contient le quartet de poids faible à convertir |
| Paramètres sorties | W (1 byte) : contient l’octet converti |
| Traitements | W=W + 0x30 |

|  |  |
| --- | --- |
| Fonctions | Routines de temporisation et pulse |
| Nom |  |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements | http://digitaldiy.io/articles/mcu-programming/assembly/55-assembly-example/114-mpasm-tutorial-liquid-crystal-display-lcd#.Wi5383mDO9I |

* + aff.asm :

|  |  |
| --- | --- |
| Fonctions | Affichage du message de calibration TBD |
| Nom | lcd\_affcalib |
| Paramètres entrée | ADCfwd  ADCref |
| Paramètres sorties | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | A | D | C | f | w | d |  | x | x | x | x |  |  |  |  |  | | A | D | C | r | e | f |  | x | x | x | x |  |  |  |  |  | |
| Traitements |  |

|  |  |
| --- | --- |
| Fonctions | Affichage du message de boot |
| Nom | lcd\_affboot |
| Paramètres entrée | c\_swversion (1byte) : Version courante du logiciel TBD : remplit au moment de la compilation ? |
| Paramètres sorties | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | S | W | R | - | P | O | W | E | R |  | m | e | t | e | r |  | | F | 8 | K | G | L |  |  |  |  |  |  |  | v | 0 | . | 1 | |
| Traitements | * v\_charpos  = 0x00 * Tant que W≠0   + Récupérer le message de boot ligne 1 (bootmsgL1)   + Afficher 1 caractère sur le LCD (lcd\_affchar)   + Incrementer v\_charpos * Positionner le curseur sur la ligne 2   + W=0x10   + Positionner le curseur du LCD (lcd\_setposcursor) * v\_charpos  = 0x00 * Tant que W≠0   + Récupérer le message de boot ligne 2 (bootmsgL2)   + Afficher 1 caractère sur le LCD (lcd\_affchar)   + Incrementer v\_charpos * Positionner le curseur sur la ligne 2, 13ème colonne   + W=0x1C   + Positionner le curseur du LCD (lcd\_setposcursor) * Afficher le caractère ‘v’   + W=’v’   + Appeler lcd\_affchar (W=’v’) * TBD : afficher la version |

|  |  |
| --- | --- |
| Fonctions | Message de boot ligne 1 du LCD |
| Nom | bootmsgL1 |
| Paramètres entrée | v\_charpos : position du caractère à retourner |
| Paramètres sorties | W (1 byte) : contient le caractère ou 0x00 si pas de caractère |
| Traitements | Zone mémoire dédiée au stockage du message de boot (ligne 1 du LCD) contenant la chaîne suivante :  « SWR-POWER meter »   * Additionner le pointeur de programme avec v\_charpos * Retourner le caractère contenu en mémoire à cette position dans W * Fin de chaîne = retourner 0x00 dans W |

|  |  |
| --- | --- |
| Fonctions | Message de boot ligne 2 du LCD |
| Nom | bootmsgL2 |
| Paramètres entrée | v\_charpos : position du caractère à retourner |
| Paramètres sorties | W (1 byte) : contient le caractère ou 0x00 si pas de caractère |
| Traitements | Zone mémoire dédiée au stockage du message de boot (ligne 2 du LCD) contenant la chaîne suivante :  « F8KGL »   * Additionner le pointeur de programme avec v\_charpos * Retourner le caractère contenu en mémoire à cette position dans W * Fin de chaîne = retourner 0x00 dans W |

|  |  |
| --- | --- |
| Fonctions | Message de version courante du logiciel |
| Nom | c\_swversion |
| Paramètres entrée | V\_charpos : position du caractère à retourner TBD |
| Paramètres sorties | W (1 byte) : contient le caractère ou 0x00 si pas de caractère |
| Traitements | Zone mémoire dédiée au stockage de la version du logiciel   * Additionner le pointeur de programme avec v\_charpos * Retourner le caractère contenu en mémoire à cette position dans W * Fin de chaîne = retourner 0x00 dans W   Cette zone devra être remplit par l’outil de compilation (TBD) |

* + lcd\_affmeas
  + lcd\_affpfwd
  + lcd\_affpref
  + lcd\_affswr
* /sw/calc
  + calc\_swr
  + calc\_power
* /sw/readadc
  + readadc\_fwd
  + readadc\_ref
* /sw/hw/
  + schéma, brd
  + définition des ports du PIC (LCD, ADC, etc...)