* 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

# 1-INTRODUCTION

# 2-MATERIEL

PIC (+sa programmation)

LCD

Ligne de mesure

ADC

Boitier

Alimentation

# 3-LOGICIEL

## 3.1-Généralités

L’architecture hardware étant fondée sur le PIC 16F628 de chez Microchip, le langage de développement du logiciel sera l’assembleur.

Le logiciel sera développé sous Linux (Ubuntu 16.04 ou Debian 8).

### 3.1.1-Application

L’application sera généré au format « .hex » pour être programmée dans la mémoire du microcontrôleur : « swr\_power\_meter\_f8kgl-Vn.m.hex »

Elle contiendra, essentiellement, 2 fonctionnalités :

* Fonctionnalité de calibration

TBD

* Fonctionnalité de mesure de la puissance transmise, de la puissance réfléchie, et le calcul du SWR

TBD

## 3.2-Outils de développement

### 3.2.1-Assembleur

Sous linux, la suite « GPUTILS », permet la compilation d’un projet développé en assembleur pour PIC.

GPUTILS est une collection d’outil pour les microcontroleurs PIC. Elle inclut :

* Gpasm : compilateur assembleur
* Gplib : compilateur assembleur permettant la génération d’une librairie
* Gplink : éditeur de lien symboliquez

Installation :

1. Désinstaller la version courante de la distribution
2. Télécharger la version 1.5.0-1 en suivant ce lien : <https://sourceforge.net/projects/gputils/files/gputils/1.5.0/gputils-1.5.0-1.tar.gz/download>
3. Installation

$ tar –xvzf gputils-1.5.0-1.tar.gz

$ cd gputils-1.5.0-1.tar.gz

$ ./configure

$make

$sudo make install

### 3.2.2-Outils de validation

Sous linux, la suite « gpsim » permet la simulation d’un code compilé par GPUTILS

Installation :

1. Désinstaller la version courante de la distribution
2. Télécharger la version 0.30.0 en suivant ce lien :

<https://sourceforge.net/projects/gpsim/files/gpsim/0.30.0/gpsim-0.30.0.tar.gz/download>

1. Installation

$ tar –xvzf gpsim-0.30.0.tar.gz

$ cd gpsim-0.30.0.tar.gz

$ ./configure

$make

$sudo make install

Utilisation :

$ gpsim –s nom\_du\_fichier.cod

1. Aller dans File->Open et choisir le fichier .stc

### 3.2.3-Versions

SVN

|  |  |
| --- | --- |
| V0.x | Implémente les fonctions LCD |
| V1.x | Implémente les fonctionsADC |
| V2.x | Implémente les fonctions de calcul |

## 3.2-Architecture

bin

ros\_metre

prj

doc

sim

hw

src

TBD

lcd

sw

readadc

calc

inc

|  |  |
| --- | --- |
| ros\_metre/bin | Contient l’ensemble des binaires produits :   * \*.a : librairie associée à un composant sw * \*.cod : simulation * \*.hex : binaire à flasher dans le PIC * \*.map : mapping mémoire * \*.cof : fichier objet résultat de la compilation * \*.lst : ? |
| ros\_metre/prj | Contient le Makefile du projet, et le point d’entrée sw (main.asm) |
| ros\_metre/doc | Documentation du projet |
| ros\_metre/src | Contient les sources du projet |
| Ros\_metre/src/hw | Contient les sources HW du projet |
| Ros\_metre/hw/sim | Contient le fichier netlist pour gpsim |
| Ros\_metre/hw/TBD | Contient les schéma, routages, gerber etc |
| Ros\_metre/sw/lcd | Composant LCD   * Driver.asm : driver bas niveau du LCD * Aff.asm : routines haut niveau d’affichage des messages * Makefile : make de la librairie LCD |
| Ros\_metre/sw/readadc | Composant ADC |
| Ros\_metre/sw/calc | Composant CALC |
| Ros\_metre/sw/inc | Include |

## 3.3-Spécifications

### 3.3.2-/prj

#### 3.3.2.1-Makefile

#### 3.3.2.2-Main.asm

|  |  |
| --- | --- |
| Fonctions | Fonction principale, point d’entrée du logiciel |
| Version | 0.1 (sauf les TBD) |
| 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)   + afficher le message de calibration (lcd\_affcalib TBD)   + Dans une boucle infinie     - lire les registres ADCfwd et ADCref     - afficher le message de mesure (lcd\_affmeas TBD) * Sinon   + Dans une boucle infinie (TBD) :     - lire les registres ADCfwd et ADCref     - calculer la puissance FWD et REF     - Calculer le SWR     - Afficher le message de mesure |

### 3.3.3/sw/inc

#### 3.3.3.1-lcd.inc

### 3.3.4-/sw/lcd

#### 3.3.4.1-driver.asm :

|  |  |
| --- | --- |
| Fonction | Initialisation du LCD |
| Version | 0.1 |
| Nom | lcd\_init |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements |  |

|  |  |
| --- | --- |
| Fonction | Affichage d’un caractère |
| Version | 0.1 |
| Nom | lcd\_affchar |
| Paramètres entrée | W(1 byte) : contient le caractère à afficher à la position courante du curseur |
| Paramètres sorties |  |
| Traitements |  |

|  |  |
| --- | --- |
| Fonction | Envoi d’une commande au LCD |
| Version | 0.1 |
| 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 |  |

|  |  |
| --- | --- |
| Fonction | Positionner le curseur du LCD |
| Nom | lcd\_setposcursor |
| Version 0.1 |  |
| 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) |

|  |  |
| --- | --- |
| Fonction | Efface le LCD |
| Version | 0.1 |
| Nom | lcd\_clear |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitement | 1. W=0x01 2. Envoi de la commande au LCD (lcd\_sendcmd) |

|  |  |
| --- | --- |
| Fonction | Positionne le curseur sur la 2ème ligne |
| Version | 0.1 |
| Nom | lcd\_setposL2 |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements | 1. W=0xC0 2. Envoi de la commande au LCD (lcd\_sendcmd) |

|  |  |
| --- | --- |
| Fonction | Conversion hexa-ASCII |
| Version | 0.1 |
| 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 |

|  |  |
| --- | --- |
| Fonction | Routines de temporisation et pulse |
| Version | 0.1 |
| Nom |  |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements | <http://digitaldiy.io/articles/mcu-programming/assembly/55-assembly-example/114-mpasm-tutorial-liquid-crystal-display-lcd#.Wi5383mDO9I> |

#### 3.3.4.2-aff.asm :

|  |  |
| --- | --- |
| Fonction | Message de version courante du logiciel |
| Version | 0.1 |
| Nom | c\_swversion |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements | Zone mémoire (5 bytes) dédiée au stockage de la version du logiciel  « Vn.m »,0x00  Cette zone de mémoire est placée au début de l’EEPROM (0x2100).  Cette zone mémoire doit se terminer par l’octet 0x00.  Cette zone mémoire est remplie par le compilateur. |

|  |  |
| --- | --- |
| Fonctions | Message de boot ligne 1 du LCD |
| Version | 0.1 |
| 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 |
| Version | 0.1 |
| 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 calibration L1 du LCD |
| Version | 0.2 |
| Nom | calibmsgL1 |
| 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 :  « ADCfwd » *avec un espace à la fin*   * 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 calibration ligne 2 du LCD |
| Version | 0.2 |
| Nom | calibmsgL2 |
| 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 :  « ADCref » *avec un espace à la fin*   * 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 |

|  |  |
| --- | --- |
| Fonction | Affichage du message de boot |
| Version | 0.1 |
| Nom | lcd\_affboot |
| Paramètres entrée | -bootmsgL1 : zone mémoire (15 bytes) contenant le message de boot ligne 1  -bootmsgL2 : zone mémoire (5 bytes) contenant le message de boot ligne 2  -c\_swversion : zone EEPROM (5bytes) contenant la version courante du logicielle |
| Paramètres sorties | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | S | W | R | - | P | O | W | E | R |  | m | e | t | e | r |  | | F | 8 | K | G | L |  |  |  |  |  |  |  | v | 0 | . | 1 | |
| Traitements | 1. v\_charpos  = 0x00 2. Afficher le message de boot ligne 1   Tant que W≠0   * + Récupérer 1 caractère du message de boot ligne 1 (bootmsgL1) dans W   + Afficher 1 caractère sur le LCD (lcd\_affchar)   + Incrementer v\_charpos  1. Positionner le curseur sur la ligne 2 :    * W=0x10    * Positionner le curseur du LCD (lcd\_setposcursor) 2. v\_charpos  = 0x00 3. Afficher le message de boot ligne 2   Tant que W≠0   * + Récupérer 1 caractère du message de boot ligne 2 (bootmsgL2)   + Afficher 1 caractère sur le LCD (lcd\_affchar)   + Incrementer v\_charpos  1. Positionner le curseur sur la ligne 2, 13ème colonne :    * W=0x1C    * Positionner le curseur du LCD (lcd\_setposcursor) 2. v\_charpos  = 0x00 3. afficher la version   Tant que W≠0   * + Récupérer le caractère en EEPROM   W ← v\_charpos  EEADDR ← W  EECON(RD) = b’1’  W ← EEDATA   * + Afficher 1 caractère sur le LCD (lcd\_affchar)   + Incrementer v\_charpos  1. TBD : afficher la version |

|  |  |
| --- | --- |
| Fonction | Affichage du message de calibration |
| Version | 0.2 |
| Nom | lcd\_affcalib |
| Paramètres entrée | -calibADCL1 : zone mémoire (7bytes) contenant le message de calibration L1  -calibADCL2 : zone mémoire (7bytes) contenant le message de calibration L1 |
| Paramètres sorties | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | A | D | C | f | w | d |  |  |  |  |  |  |  |  |  |  | | A | D | C | r | e | f |  |  |  |  |  |  |  |  |  |  | |
| Traitements | 1. v\_charpos  = 0x00 2. Afficher le message de calibration ligne 1   Tant que W≠0   * + Récupérer 1 caractère du message de calibration ligne 1 (calibmsgL1) dans W   + Afficher 1 caractère sur le LCD (lcd\_affchar)   + Incrementer v\_charpos  1. Positionner le curseur sur la ligne 2 :    * W=0x10    * Positionner le curseur du LCD (lcd\_setposcursor) 2. v\_charpos  = 0x00 3. Afficher le message de calibration ligne 2   Tant que W≠0   * + Récupérer 1 caractère du message de calibration ligne 2 (calibmsgL2)   + Afficher 1 caractère sur le LCD (lcd\_affchar)   + Incrementer v\_charpos |

|  |  |
| --- | --- |
| Fonction | Affichage de la mesure de calibration TBD |
| Version | 1.0 |
| Nom | lcd\_affmeas |
| Paramètres entrée |  |
| Paramètres sorties | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  | x | x | x | x |  |  |  |  |  | |  |  |  |  |  |  |  | y | y | y | y |  |  |  |  |  | |
| Traitements |  |

TBD

lcd\_affpfwd

lcd\_affpref

lcd\_affswr

### 3.2.4-/sw/calc

#### 3.2.4.1-calc\_swr

#### 3.2.4.2-calc\_power

### 3.2.5-/sw/readadc

#### 3.2.5.1-readadc\_fwd

|  |  |
| --- | --- |
| Fonction | TBD |
| Version | 1.0 |
| Nom |  |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements |  |

#### 3.2.5.2-readadc\_ref

|  |  |
| --- | --- |
| Fonction | TBD |
| Version | 1.0 |
| Nom |  |
| Paramètres entrée |  |
| Paramètres sorties |  |
| Traitements |  |

Compilation

Plan mémoire

### 3.2.6-/sw/hw/

#### 3.2.6.1-schéma, brd

#### 3.2.6.2-définition des ports du PIC (LCD, ADC, etc...)