Test du code VHDL

Contents

Cette page décrit brièvement l'historique des tests basiques du code VHDL developpé par Marc. Utilisation de la librairie standard fournie par FTDI. Les fonctions C serviront de base pour les **'Producteurs'** et les **'Filtres'** nécéssaires au fonctionnement de NARVAL.

1 Installation des drivers ftdi

Ce chapitre a pour but de conserver une trace sur l'installation des drivers FTDI, disponible sous forme de librairie à l'adresse suivante: http://www.ftdichip.com/Drivers/D2XX.htm.

1.1 Installation des librairies

- création d'un repertoire *ftdi/* dans lequel on copie les divers; pour linux 32 bits:

  ```
  wget http://www.ftdichip.com/Drivers/D2XX/Linux/libftd2xxy.y.yy.tar.gz;
  ```

- décompresssions des fichiers:

  ```
  gunzip libftd2xxy.y.yy.tar.gz \\
  tar -xvf libftd2xx0.4.12.tar \\
  ```

- se connecter en tant qu'administrateur
- `cp libftd2xx.so.0.4.12 /usr/local/lib`
- `ln -s libftd2xx.so.0.4.12 libftd2xx.so`
- `cd /usr/lib`
- `ln -s /usr/local/lib/libftd2xx.so.0.4.12 libftd2xx.so`
- dans le fichier */etc/fstab*, ajouter le ligne suivante:

  ```
  none /proc/bus/usb usbfs defaults,devmode=0666 0 0
  ```

- `/mount -a`

1.2 Droits Utilisateurs

Dans l'état actuel de m'installation, seul l'administrateur est en droit d'utiliser le liaison USB/FTDI. La procédure suivante permet l'utilisation de ce driver pour tout autre utilisateur.

- Ajout du groupe **ftdi**: `addgroup ftdi`
- Ajouter dans ce groupe les utilisateurs devant accéder au périphérique en editant le fichier */etc/group*.
- créer un fichier */etc/udev/rules.d/026_ftdi.rules* tel que:

  ```
  #FTDI device
  SYSFS{idVendor}=="0403", SYSFS{idProduct}=="6001", MODE="0660", GROUP="ftdi"
  ```

- redémarrer le périphérique **udev** ou redémarrer.

2 Installation du module usb/gpib

Lors des differents tests q effectue, en plus des cartes FTDI, nous aurons a controler via une liaison GPIB:

- les alimentations des detecteurs et de l'acquisition
- le controle du laser

Solution d'une interface USB/GPIB dont voici la procedure d'installation. Installation des librairiesm code source et outils fxload;

2.1 Installation des librairies

Les documentations liee a l'installation et l'utilisation de ce peripheriaue sont:

- http://linux-gpib.sourceforge.net/
- http://linux-hotplug.sourceforge.net/
- http://linux-gpib.sourceforge.net/firmware/

Sous distribution Debian, il y a la possiblite d'installer directement les librairies via la commande:

- apt-get install gpib-modules-source libgpib-bin libgpib0 libgpib0-dev (

  `http://packages.debian.org/search?keywords=gpib&searchon=names&suite=stable&section=all`

)
- apt-get install fxload (

  `http://packages.debian.org/search?suite=stable&section=all&arch=any&searchon=names&keywords=`

)

A partir des codes sources (en particulier sous SL5.2), il faut:

- créer un repertoire GPIB (au meme niveau que les drivers FTDI dans notre exemple)
  mkdir GPIB

- cd GPIB
- on copie des drivers disponibles sur la page:

  `http://sourceforge.net/project/showfiles.php?group_id=42378`

  ;
- copie des firmaware disponible sur le lien suivant:

  `http://linux-gpib.sourceforge.net/firmware/`
- copie du module fxload sur:

  `http://sourceforge.net/project/showfiles.php?group_id=17679`
- décompresssions de tous les fichiers
- dans le repertoire de fxload,: make make install (mode administrateur)
- dans le repertoire contenant le code source linux-gpib: ./configure make make install (mode administrateur)
- en mode adminstrateur, copier les firmware dans le repertoire suivant:

```
cp measat_releaseX1.8.hex /usr/share/usb/agilent_82357a/
cp 82357a_fw.hex /usr/share/usb/agilent_82357a/
```

- `fxload -t fx2 -D /proc/bus/usb/001/003 -I measat_releaseX1.8.hex`

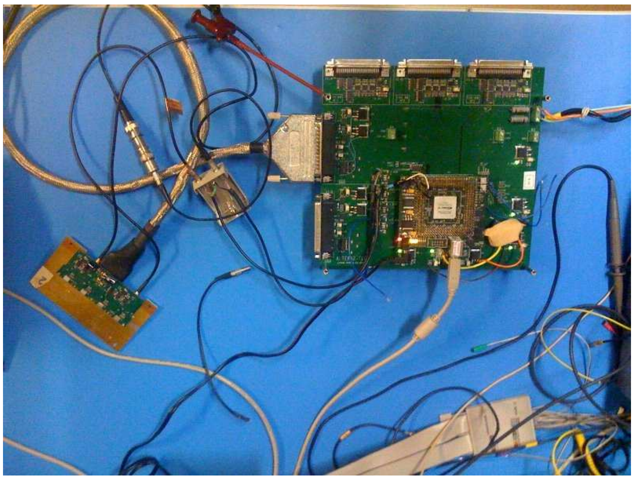
en cours ...

3 Historique des tests

3.1 Remise en marche d'une carte Altera2

Jacques et Alexandre ont remis en route l'une des cartes ALTERA2 utilisées lors du test sous faisceau du CERN selon le protocole présenté figure ??. Sont utilisées:

- une carte ALTERA2 des tests sous faisceau du CERN 2008;
- une carte frontale VA sans senseurs (état du bonding très limite)
- connection via USB/EtherneExtander/USB



Quelques mesures ont été éffectuées à partir du code VHDL d'Albert et le soft C++ modifié. Pour petit rappel, à cause d'un cablage défectueux du hardware, le décodage des ADC 13 et 24 est différent:

- ADC 13: numérisation en entier *'décalé'* (i.e. $+2^{13}$);
- ADC 24: numérisation en entier non signé;

De ces premiers tests, un problème sur l'ADC 1 à été observé. Celui-ci semble saturer lorsque l'on connecte la lecture des VA. l'ADC 2 semble normal.

3.2 Premiers tests du nouveau code vhdl

3.2.1 Descritpion du code vhdl

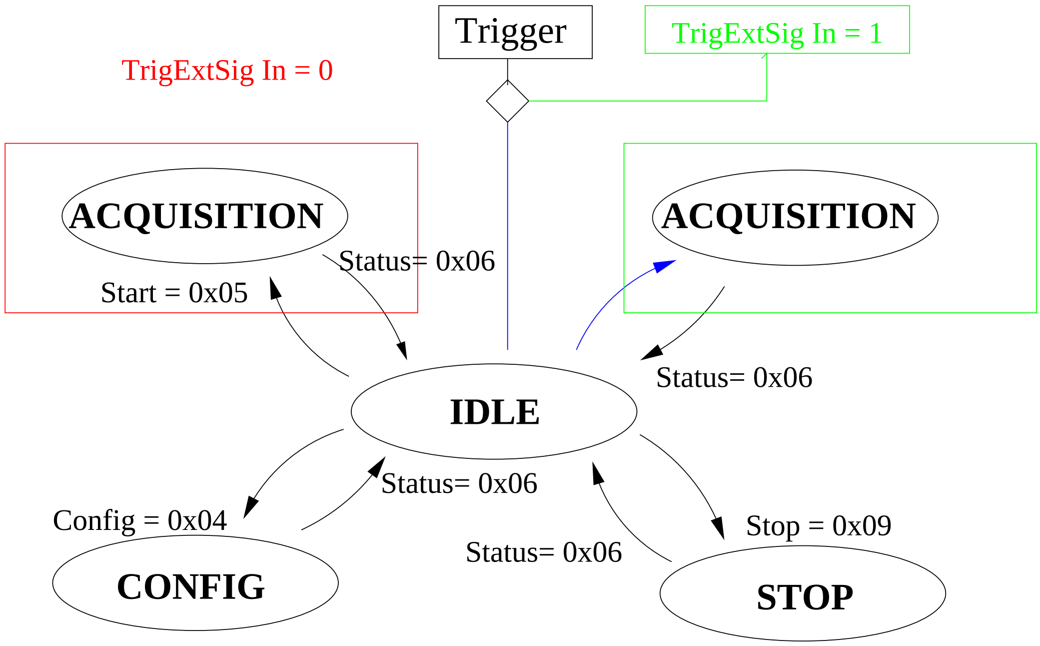
Le code VHDL testé est basé sur le design pattern d'Albert. (discuté avec Marc des diagrammes). Les fonctions C++ seront appelées génériquement **VAprogC**.

L'acquisition est réalisée selon les étapes suivantes (le premier bloc étant optionnel):

- l'utilisateur envoie la commande *'Configuration'*;
- le VHDL se met en mode *'reception de la configuration'* et stocke en mémoire les informations envoyées par VAprogC telles que, le *'hold'* en 3 octets, l'adresse de la carte et la configuration de l'ALTERA2.

- l'utilisateur envoie la commande *'Start'* pour démarer l'acquisition;
- le **VAprogC** envoie un signal *'trigger'*;
- le VHDL déclenche la séquence de lecture des VA1;
- les données sont stockées dans une FIFO du FPGA qui envoie l'état *'données présentes'* dans la FIFO de l'USB ;
- le **VAprogC** récupère ce mot et déclenche la lecture des $2n$ blocs de 128 octets, n étant le nombre de VA1 à lire. les données d'un VA1 arrive donc en deux blocs de 128 octets, le premier contenant tous les MSB, et le second les LSB;
- une fois tous les blocs récupérés par **VAprogC**, les données sont remises en forme et enregistrées dans un fichiers texte.
- **VAprogC** envoie la commande d'un nouveau *'trigger'*;
- cette séquence est répétée le nombre de fois voulue par l'utilisateur. A la fin, le **VAprogC** envoie la commande *'Stop'* (facultatif).

Le diagramme d'état du code VHDL est représenté figure **??**. Sur ce diagramme sont représentés deux modes d'acquisition: l'un à partir d'un *'trigger soft'* commandé par le **VAProgC**, le second à partir d'un trigger externe.



3.2.2 Résultats