

Linux Embarqué

C en compilation croisée

avec Eclipse Kepler

Travaux pratiques

**Système Linux embarqué Debian 7 (Wheezy) sur la carte
BeagleBone (White/Black)
basée sur microcontrôleur
Texas instruments AM335x (ARM Cortex A8)**

Matière	Microcontrôleur - Linux Embarqué
Support	Travaux pratiques - Introduction à Linux Embarqué
Prérequis	Cours & TD Introduction à Linux Embarqué– Yves Auffret
Année	2013/2014
Durée	7 heures
Version	1.00– 2014
Enseignant	Yves Auffret

TP 1

Durée 3 heures

**Acquisition de données avec le
convertisseur analogique/numérique et
publication des résultats sur le serveur web
Apache**

Matériel nécessaire :

- Carte BeagleBone (white) rev. >=A6 <http://beagleboard.org/Products/BeagleBone>
ou
- Carte BeagleBone (black) rev. >=A6 <http://beagleboard.org/Products/BeagleBone+Black/>
- Carte MicroSD de 4GO
- Câble miniUSB-USB ou une alimentation DC 5V
- Un ordinateur (Windows, Linux ou Mac OS X) connecté au réseau ISEN
- Câble RJ45 permettant de connecter la carte BeagleBone au réseau ISEN
- Lecteur de carte mémoire MicroSD pour l'ordinateur

L'objectif du TP est de réaliser un système d'acquisition de données avec le convertisseur analogique/numérique et de publier les résultats sur le serveur web Apache de la carte Beaglebone.

1) Acquisition des données

La fonction acquisition de données consistera à enregistrer dans un fichier texte toutes les secondes les résultats des conversions des voies AN1, AN2, AN3 et AN4.

Le format proposé est le suivant :

Date Heure AN1 AN2 AN3 AN4

Date Heure AN1 AN2 AN3 AN4

....

Date Heure AN1 AN2 AN3 AN4

2) Publication des résultats sur le serveur web Apache

Configurer le serveur web Apache afin d'afficher les données à partir d'un navigateur web.

3) Compte rendu

Rédiger un compte rendu de TP incluant les réponses aux questions posées ainsi que les sources des logiciels développés.

IMPORTANT

Ne pas oublier de préparer le prochain TP

TP 2

Durée 4 heures

Commande d'un servomoteur avec un signal PWM

Matériel nécessaire :

- Carte BeagleBone (white) rev. >=A6 <http://beagleboard.org/Products/BeagleBone>
- ou
- Carte BeagleBone (black) rev. >=A6 <http://beagleboard.org/Products/BeagleBone+Black/>
- Carte MicroSD de 4GO
- Câble miniUSB-USB ou une alimentation DC 5V
- Un ordinateur (Windows, Linux ou Mac OS X) connecté au réseau ISEN
- Câble RJ45 permettant de connecter la carte BeagleBone au réseau ISEN
- Lecteur de carte mémoire MicroSD pour l'ordinateur
- Oscilloscope

L'objectif du TP est de contrôler directement un servomoteur par la carte BeagleBone, deux méthodes seront utilisées afin de les comparer :

- Génération de signaux PWM par logiciel
- Génération de signaux PWM par le contrôleur intégré AM335x (ARM Cortex A8)

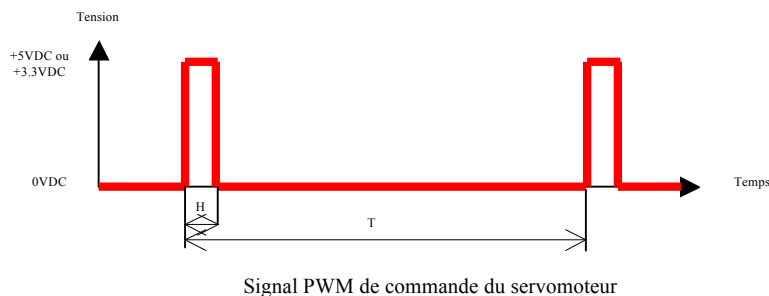
Les servomoteurs sont très utilisés en robotique comme actionneurs, l'utilisation dans ce TP d'un servomoteur de modélisme est pour illustrer le concept en sachant que d'un point de vue logiciel le principe est identique pour piloter, par exemple, un servomoteur PWM plus puissant ou un moteur brushless avec un contrôleur PWM.

!!! Attention Utiliser uniquement les sorties VDD_5V accessibles sur les connecteurs pour alimenter le servomoteur.

!!! Attention Avant de connecter le servomoteur, vérifier votre câblage et la conformité des signaux générés avec un oscilloscope, pour éviter que le servomoteur ne se retrouve en butée et finisse par se détériorer.

!!! Attention Ne pas insérer la pointe de la sonde de l'oscilloscope dans les connecteurs ou dans la plaque d'expérimentation, cela détériorerait les contacts et la pointe de la sonde. Il est nécessaire d'utiliser un des fils fournis entre un connecteur et la sonde ou entre la plaque d'expérimentation et la sonde.

Le signal PWM est défini de la façon suivante :



L'angle du servomoteur est compris entre -45° et $+45^\circ$, H représente la durée à l'état haut, avec la relation linéaire suivante :

- -45° correspond à une durée à l'état haut H de 1 ms
- 0° correspond à une durée à l'état haut H de 1.5 ms
- $+45^\circ$ correspond à une durée à l'état haut H de 2 ms

La période T du signal doit être de 20ms

Câblage du connecteur 3 points du servomoteur Futaba S3003 :

- noir = masse
- rouge = alimentation +5VDC
- blanc = entrée signal PWM compatible 5VDC et 3.3VDC

4) Commande du servomoteur par logiciel

En utilisant un GPIO de votre choix, écrire la procédure `cmd_servo_soft(angle)` en langage C permettant de gérer le servomoteur par logiciel.

Vérifier le signal PWM avec un oscilloscope.

- Que constatez-vous ?
- Que se passe-t-il lorsque la charge du système Linux varie ?
- Pouvez-vous améliorer le respect des contraintes temporelles ?

5) Commande du servomoteur par le contrôleur PWM

En utilisant une sortie de votre choix sur le contrôleur PWM, écrire la procédure `cmd_servo_hard(angle)` en langage C permettant de gérer le servomoteur avec le contrôleur PWM intégré au AM335x (ARM Cortex A8).

Contrôler le signal PWM avec un oscilloscope.

- Que constatez-vous ?
- Que se passe-t-il lorsque la charge du système Linux varie ?

Réaliser un programme permettant de faire varier le signal PWM de la façon suivante :

- positionner le servomoteur à -10° pendant 5 secondes
- positionner le servomoteur à 0° pendant 5 secondes
- positionner le servomoteur à $+10^\circ$ pendant 5 secondes

Vérifier le signal PWM avec un oscilloscope.

6) Compte rendu

Rédiger un compte rendu de TP incluant les réponses aux questions posées ainsi que les sources des logiciels développés.