

Jacob Fontaine
Philippe Spino

Spip2401
Fonj

Rapport App2

Présente à :
JEAN LAVOIE

January 30, 2018

Contents

1	Introduction.	2
2	Utilisation des ressources matérielles.	2
3	Algorithmes de gestion des acquisitions de données.	3
3.1	Numérique	3
3.2	Analogique	4
4	Synchronisation des tâches CPU.	4
5	Conception du montage prototype.	4
6	Méthode de validation des fréquences d'échantillonnage.	4
7	Optimisation du prototype.	4
8	Analyse de performances des Codes de Métrologie Numérique.	4
9	Conclusion.	4

1 Introduction.

Ce présent rapport a pour but de présenter les éléments de conception du système d'acquisition de données(SAD) développé par notre équipe. Le SAD a été conçu à l'aide d'un processeur embarqué LPC1768, soit la coquille de base du SAD. Le contenu de ce rapport explique les ressources mémoire et du processeur utilisées, les algorithmes de gestion des acquisitions, la synchronisation des tâches utilisées, une analyse des problèmes de conception et de déverminage lors du montage du prototype, une description des méthodes concernant la validation des fréquences d'échantillonnages des données et une analyse du prototype le plus optimal. De plus, une discussion sur les temps d'exécution des codes base pour un inclinomètre. à la demande de la compagnie metrologie numérique.

2 Utilisation des ressources matérielles.

Pour ce présent prototype deux LCP1768 sont utilisés. Un des microcontrôleur est programmé pour émettre de façon aléatoire des données soit numériquement ou bien analogiquement. Il est nommé DataSpammer. L'autre microcontrôleur utilise l'interface principale du SAD. Le DataSpammer, est réglé avec une fréquence d'opération mobile, c'est-à-dire qu'elle peut être changée selon l'utilisateur. Pour le montage, nous avons 2 mbed qui sont branchés ensemble et le DataSpammer envoie ses données aléatoires via 2 port GPIO et la fréquence d'opération passe de 10ms à 500ms. Sur le microcontrôleur qui a l'interface primaire, les led 1 et 2 servent à indiquer un changement de voltage sur les pattes analogiques p19 et p20. Les led 3 et 4 servent à indiquer un événement numérique. Les pattes p21 et p22 sont utilisées.

L'interface principale utilise 4 tâches pour faire la gestion des acquisitions, une tâche pour chaque type d'entrée, une tâche pour l'horloge temps-réel(RTC) et une tâche pour la gestion de la propagation. Pour propager les données acquises, une structure, nommée Event, est utilisée. Cette structure contient un byte non signé, un entier de 32bit et une struct temporelle. Pour propager les Event, une file globale, qui contient des Event, de type First In First Out(FIFO) est utilisée. Pour la gestion de l'accès mémoire de cette file, un mutex global est utilisé. La file est utilisée par les deux tâches d'acquisitions numérique et analogique. Par la suite, pour l'acquisition du temps lors d'une capture de données, une file globale FIFO est utilisée pour une demande du temps. un mutex est aussi utilisé pour la gestion de l'accès à cette file.

3 Algorithmes de gestion des acquisitions de données.

L'algorithme développé pour l'acquisition des données est divisé en 4 tâches différents. Numérique, Analogique et le RTC. La tâche pour l'acquisition numérique étant en priorisation haute, l'analogique en priorisation moyenne et le reste, soit la tâche pour le RTC, en priorisation basse.

3.1 Numérique

La tâche pour l'acquisition des données numérique vérifie s'il y a un changement de bit sur le front montant des changements des bits. Le temps pris pour qu'un bit puisse changer est de l'ordre des 50 ms. La fréquence d'opération de cette algorithmes est 100ms. Pour ce faire, une tâche::wait est utilisée. cela permet d'être assuré que la tâche va être à une fréquence de 100ms. Une boucle itérative sur 20 tours regarde les valeurs des pattes d'entrées digitales

à toutes les 5ms, lorsque les valeurs rester inchangé après une itération, un compteur est incrémenté. Si dès que le compteur atteint 10($10 * 5\text{ms} = 50\text{ms}$) un Event est créé avec l'acquisition et on l'insère dans la file des Event. Prio

3.2 Analogique

4 Synchronisation des tâches CPU.

Pour la synchronisation des ressources et des tâches, des mutexs sont utilisés. Le code opère à l'aide d'une liste de type FIFO.

5 Conception du montage prototype.

6 Méthode de validation des fréquences d'échantillonnage.

7 Optimisation du prototype.

8 Analyse de performances des Codes de Métrologie Numérique.

9 Conclusion.