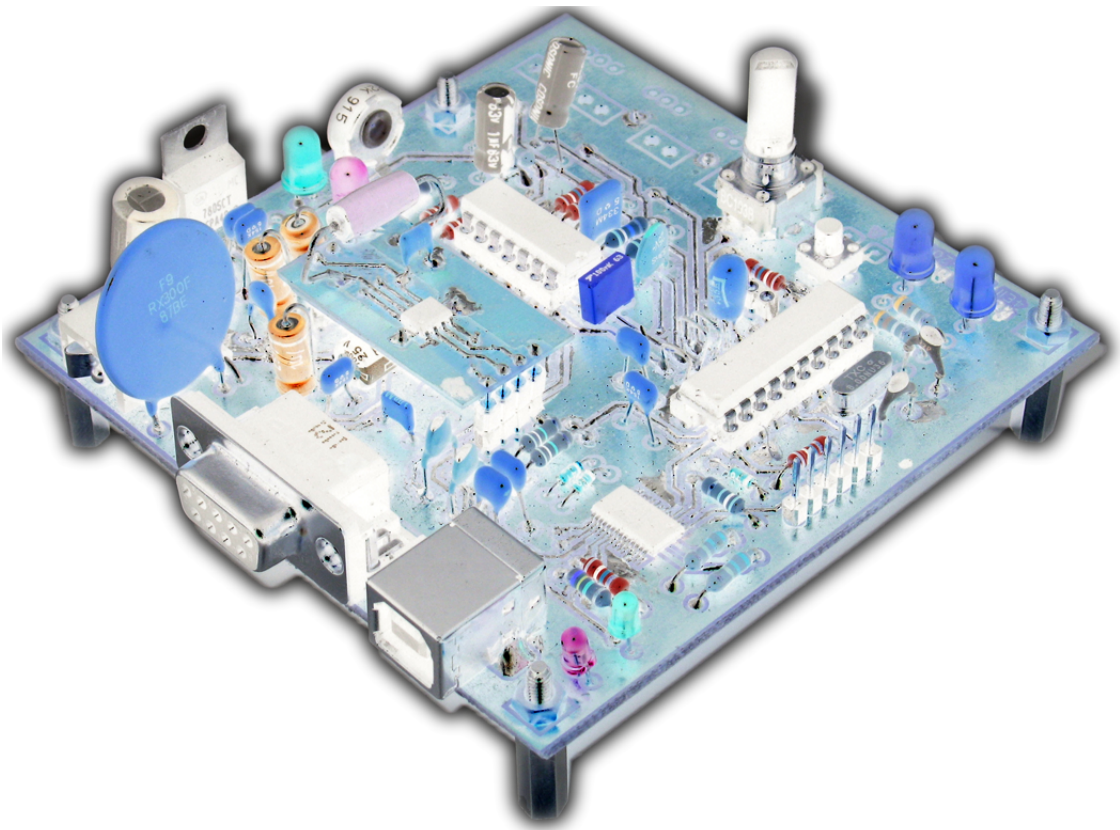


PROJET PLURIDISCIPLINAIRE

Solution Numérique



Xavier GALZIN, Stanislas BERTRAND, Romain DESILLE, Frédéric MESLIN

23 mai 2012

Table des matières

1	Communication	3
1.1	Choix de la liaison USB	3
1.2	Utilisation d'un contrôleur USB - Série	4
1.3	Élaboration d'un protocole	4
1.3.1	Configuration générale	5
1.3.2	Différentes trames	5
1.4	Implantation du protocole	5
2	Interface	5
3	Changement de Puissance	6
3.1	Causes du changement	6
3.2	Choix du nouveau pont en H	6
3.3	Fonctionnement du L6203	7
4	Correction Numérique	7
5	Répartition des gains	7
5.1	Correcteur Analogique	7
5.2	Correcteur Numérique	7
6	Gimmicks	7

Introduction

Dans nos précédents rapports, nous avons détaillé le problème de l'asservissement en position du mobile et nous avons déterminé les paramètres d'un modèle analogique de correcteur. Par la suite, nous avons mis en place la théorie d'un protocole de communication série afin de faciliter les réglages et les mesures en temps réel. Dans le troisième rapport, nous avons décrit l'avancement de la partie analogique du projet, de la conception de la maquette jusqu'à sa réalisation et ses tests.

Dans ce dernier rapport, nous allons nous concentrer sur la correction numérique. Nous commencerons par définir les caractéristiques du protocole série que nous avons décidé d'utiliser en lieu et place de celui défini dans le rapport de MCSE puis nous en expliciterons la programmation dans le microcontrôleur. Ensuite, nous aborderons brièvement la question de l'interface graphique mise en place et nous reviendrons également sur le changement intervenu dans la partie puissance avant de nous intéresser à la correction numérique à proprement parler. Enfin, nous clarifierons la question de la répartition des gains, en analogique comme en numérique, avant de conclure globalement sur le projet tout en proposant des possibilités d'améliorations que nous n'aurons malheureusement pas le temps de mettre en place.

1 Communication

La partie communication du projet peut paraître secondaire quand on considère l'application dans sa globalité. Pour quelles raisons un luminaire aurait-il besoin de communiquer avec un dispositif informatique ? L'intérêt semble limité ...

Dans les faits, cette fonctionnalité a été implémentée dans une optique d'assistance au développement. Elle s'est avérée utile pour mettre au point l'asservissement numérique et faciliter le réglage des coefficients de la fonction de transfert du correcteur. Elle permet aussi désormais de collecter diverses informations liées à l'automatique pour analyser la qualité (stabilité) de l'asservissement. On imagine que cette liaison série ne sera pas intégrée dans le produit final, sauf si les spécifications devaient évoluer vers un appareil nécessitant de la connectivité. On peut penser à l'intégration dans un réseau domotique via des technologies Zigbee ou même au travers le Wifi.

La communication a aussi un intérêt pédagogique : nous avons fait le choix d'établir un protocole dédié sur une liaison de type série prise en charge par une connection USB. Nous avons préféré cette solution à celle présentée dans la partie MCSE du projet pour plusieurs raisons qui vont être détaillées.

1.1 Choix de la liaison USB

Dans l'univers de la communication péri-informatique, la liaison USB est figure souveraine. La première version de cette spécification a vu le jour en 1996 dans l'objectif de remplacer progressivement les anciennes connectiques lentes et incompatibles entre elles.

Ce document normalise à la fois :

- un protocole complet de communication série maître vers esclaves
- le matériel nécessaire pour faire transiter l'information (cable, prises ...)
- toutes les caractéristiques électriques et mécaniques associées
- des classes de pilotes standards couvrant un nombre important d'applications génériques.

Notre application étant destinée à communiquer avec un ordinateur et l'USB fournissant une classe générique de communication série équivalente à un port COM virtuel, nous avons choisi d'opter pour cette solution.



FIGURE 1: Logo USB 2.0

1.2 Utilisation d'un contrôleur USB - Série

Afin de bénéficier de la connectique USB sans entrer dans les détails de la sous-couche de communication, nécessitant des cycles de développement lourds, deux solutions s'offraient à nous :

1. Choisir un micro-contrôleur disposant de fonctionnalités USB
2. Utiliser un contrôleur USB externe

La première solution est la plus polyvalente car elle permet d'implanter n'importe quelle classe de périphérique dans le programme du micro-contrôleur en se basant sur une bibliothèque USB fournie par le constructeur. Ceci nécessite une étude approfondie de la documentation associée et la rédaction de descripteurs USBs. Etant donné l'aspect auxiliaire de la communication, ce surcroît de développement a été écarté.

La seconde solution impliquant l'ajout d'un composant externe, au détriment du coût, a été retenue pour l'aisance de développement qu'elle apporte. Cette solution clef en main convertit une liaison série TTL en provenance du micro-contrôleur en liaison USB de manière presque transparente.

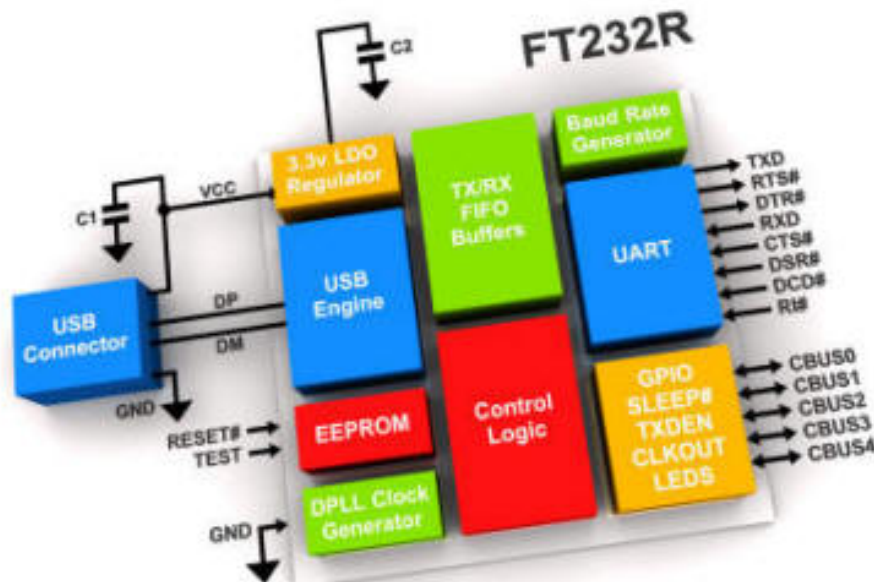


FIGURE 2: Diagramme interne du FT232

Le contrôleur circuit qui a été sélectionné est le FT232R de FTDI. Ce composant a été très simple à intégrer sur la carte du projet car il ne nécessite qu'une alimentation 5v et la liaison série pour fonctionner. De plus, il configure ses paramètres de transmission automatiquement en fonction de ceux communiqués par l'ordinateur à l'ouverture du port. Pour résumer ce composant est facile d'utilisation à défaut d'être peu coûteux.

1.3 Élaboration d'un protocole

Dans le rapport de MCSE, nous avons présenté un protocole basique, exploitant un format de trames fixe de 3 octets consécutifs. Le premier contenant l'index de la commande et les deux suivants le corps de la donnée envoyée. Nous nous sommes rendus compte que ce protocole ne couvrait pas tous nos besoins et en particulier la possibilité de transmettre des messages d'erreurs ou de log.

Nous avons donc programmé un autre protocole, plus évolué et plus évolutif. Celui ci est capable de transmettre plusieurs types de trames à longueur variable et d'inclure des codes de gestion d'erreur de transmission (au besoin).

1.3.1 Configuration générale

Les paramètres généraux utilisés dans la transmission série sont les suivants :

Taille donnée 8 bits

Parité Aucune parité

Bits de stops 1 seul

Baudrate variable de 9600 à 115200

Le Baudrate est variable car le micro-contrôleur dispose d'un module capable de déterminer automatiquement la vitesse de transmission de la liaison. La seule condition à respecter est l'émission du caractère ASCII "U" avant toute communication. Ce caractère correspond à un motif de niveaux logiques hauts et bas alternés qui permettent au périphérique série du micro-contrôleur de calibrer son registre prescaler pour générer le bon Baudrate.

1.3.2 Différentes trames

Les différentes trames disponibles sont les suivantes :

- Trame de contrôle :

typ	r/w	chr	ad4	ad3	ad2	ad1	ad0
0	0	0	x	x	x	x	x

typ : type de trame (0 : données — 1 : contrôle)

r/w : type de transfert (0 : lecture — 1 : écriture)

chr : type de données envoyées (0 : entiers 16 bits — 1 : chaîne de caractères)

ad4 ... ad0 destination de la donnée (entière) concernée (0 - 32)

- Trame de donnée entière :

MSB :	typ	b06	b05	b04	b03	b02	b01	b00
	0	0	0	0	d15	d14	d13	d12
MSB :	typ	b06	b05	b04	b03	b02	b01	b00
	0	0	0	0	d11	d10	d09	d08
LSB :	typ	b06	b05	b04	b03	b02	b01	b00
	0	0	0	0	d07	d06	d05	d04
LSB :	typ	b06	b05	b04	b03	b02	b01	b00
	0	0	0	0	d03	d02	d01	d00

typ : type de trame (0 : données — 1 : contrôle)

d15 ... d00 valeur de la donnée 16 bits

- Trame de chaîne de caractères :

typ	r/w	log	ad4	ad3	ad2	ad1	ad0
0	ch6	ch5	ch4	ch3	ch2	ch1	ch0

typ : type de trame (0 : données — 1 : contrôle)

ch6 ... ch0 caractère table ASCII standard

Pour terminer l'envoi d'une chaîne, il suffit d'envoyer une trame ne contenant que des zéros. Ceci correspondant à l'envoi du caractère ASCII de terminaison.

1.4 Implantation du protocole

2 Interface

Romain

3 Changement de Puissance

3.1 Causes du changement

Au cours de nos tests pour valider le fonctionnement de nos correcteurs analogiques et numériques, nous avons essayé de faire léviter le mobile à la distance de 1,5 cm. Nous avons eu du mal à y parvenir car notre pont en H chauffait de manière trop importante et la dissipation thermique était tout juste suffisante. La correction en résultant était donc assez instable bien que fonctionnelle.

Nous avons donc voulu améliorer ce point en rajoutant un radiateur à notre pont en H. Cependant, ce dernier étant un composant CMS, il n'était pas possible d'ajouter facilement le radiateur. Nous avons alors tenté de coller le radiateur au plan de masse du dessous du circuit de puissance par le biais d'une grosse pastille d'étain. Malheureusement, notre pont en H CMS s'est dessoudé lors de l'opération.

Nous avons alors tenté de le ressouder en utilisant le four à refusion. Bien que cela ait fonctionné, il semble que la conduction thermique ait été très largement amoindrie. Il est également possible que le pont en H ait été endommagé lors de la seconde utilisation du four à refusion. C'est à ce moment que nous avons décidé de changer de pont en H, optant pour un composant Multiwatt afin de nous fournir une plus grande marge de manœuvre pour l'asservissement en augmentant le courant maximum utilisable.

3.2 Choix du nouveau pont en H

Après consultation du catalogue Farnell, nous avons choisi un L6203 de ST Microelectronics. Bien qu'un peu plus cher (7.44 € en unitaire), ce composant fonctionne de manière similaire à l'A4950 : on peut choisir le sens de parcours du courant dans le pont par le biais des mêmes entrées qu'avec l'A4950. Cependant, ce composant nécessite des condensateurs de Bootstrap (1 par sortie) et supporte jusqu'à 5 A au lieu des 3,5 A du A4950. De plus, son boîtier Multiwatt nous assure une dissipation thermique largement meilleure qu'avant en plus de nous permettre de rajouter facilement un radiateur. En outre, le $R_{DS(ON)}$ de ce pont est de $0,3\ \Omega$ au lieu des $1,3\ \Omega$ de l'A4950 : on chauffera donc beaucoup moins tout en dissipant mieux, augmentant ainsi sensiblement notre marge de manœuvre pour l'asservissement. Enfin, le L6203 dispose également d'une protection en courant mais n'implémente plus de protection en courant. Il faudrait rajouter un comparateur sur la sortie V_{Sense} du pont en H, ce que nous ne ferons pas pour des raisons de simplifications et à cause du fait que les alimentations ne peuvent pas fournir de courant supérieur à ce que peut supporter notre pont. Ci dessous, le schéma bloc du L6203.

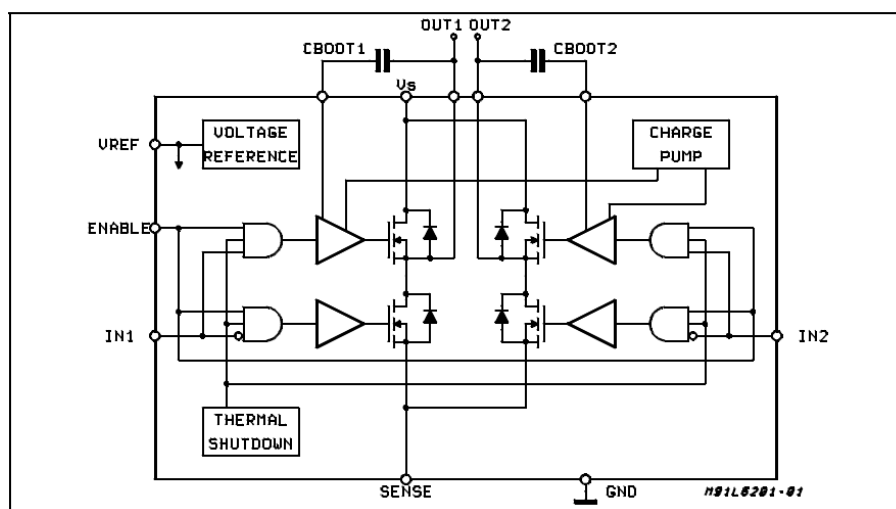


FIGURE 3: Schéma bloc interne du L6203

3.3 Fonctionnement du L6203

Après quelques mésaventures lors du routage (inversion de l’empreinte du pont en H), nous avons pu tester notre nouveau circuit de puissance. Celui-ci s’est avéré meilleure que le précédent, fournissant un asservissement plus stable en raison de sa plus grande tolérance au courant de plus grandes valeurs. La consommation a également été diminuée (entre 0,4 et 0,5 A) grâce à la baisse significative du $R_{DS(on)}$. Il aurait été possible d’optimiser la consommation au détriment de la position en amenant le mobile de sorte que la répulsion du champ magnétique soit compensée par le poids du mobile.

4 Correction Numérique

Difference avec le rapport d’auto -i Foh vers Zoh -i Fréquence de travail

Questions / causes probables : -i est-ce que le gain de l’avance de phase compte (analogique) ?
-i mauvais correcteur -i mauvais modèle -i mauvais calcul -i saturation, écretage, arrondis

Utilisation d’un outils : -i suivit des calculs via la série (diff, avance de phase, sortie)

dernière seance -i 1 gain du montage différentiel (3.5) $(1+(2*R1)/R2) == i / 2$ à cause des capteurs $== i 1,75$ -i 2 validation avec matlab de la répartition du gain (av ph, num) -i 3 validation avec l’asservissement numerique du mercredi du solex -i 4 essaye d’amélioration de l’asservissement numérique

5 Répartition des gains

Xavier - Fred

5.1 Correcteur Analogique

5.2 Correcteur Numérique

6 Gimmicks

En plus des asservissements numérique et analogiques, nous avons mis en place trois fonctionnalités additionnelles :

- La première est d’ordre pratique. En temps normal, dès que l’on cesse d’asservir, le mobile chute. Cela provoque un bruit plutôt désagréable. Nous avons donc implémenté une fonctionnalité qui amène le mobile à se coller à la bobine lorsque l’on quitte l’application graphique.

- Seconde fonctionnalité : tant que l’on est en analogique, on acquiert les mesures via la liaison série. Il est possible d’exporter ces mesures au format CSV afin de générer des courbes pour y suivre l’évolution de la position du mobile.

- La dernière est d’ordre artistique : lorsque le mobile vient se coller sur la bobine, on joue un air de musique connu¹ est faisant varier la fréquence de la PWM.

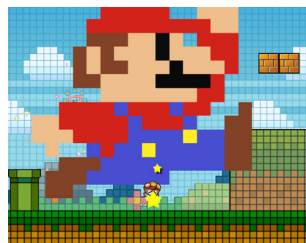


FIGURE 4: Un air connu ?

1. <http://www.youtube.com/watch?v=Kh2AWswAMvw>

Conclusion - Evolution

Xavier