RAPPORT DE PROJET M4100

BENROUIJAL IMADE – BELAMRI AHMED – BERDA SIMON

BRUNEL MICKAËL – KALOUSTIAN REMY – KUMAR RAPHAËL

TABLE DES MATIÈRES

1 INTRODUCTION

Dans le cadre du module M4100 Dev d'application et agile , nous avons à réaliser un projet informatique en utilisant des outils informatiques que nous avons acquis lors des semestres précédents.

Ce projet va nous permettre d'appliquer nos connaissances en informatique et d'en acquérir de nouvelles dans le domaine de l'embarqué et des méthodes agiles dans un cadre concret ; ce qui nous apparaît très intéressant.

Un aspect très positif de ce projet est d'apprendre à utiliser de nouveaux systèmes comme la SAM4S ou encore de nouvelles méthodes de développement comme les méthodes agiles.

Enfin, le fait de fusionner nos compétences dans deux domaines différents semble très formateur, puisque nous allons évoluer dans des environnements pluridisciplinaires au cours de nos études à venir et de notre carrière professionnelle.

2 UNE PREMIÈRE APPROCHE DE LA CARTE

Suite à la formation du groupe, ont émergé deux trinômes, l'un ayant le rôle de s'occuper de la luminance et de la colorimétrie , l'autre recevant la lourde de tâche de faire fonctionner le servomoteur.

Une fois la distribution installée, deux questions se sont posées :

— Comment flasher le nouveau système sur la carte ?

— Comment réaliser nos tests et nos mocks ?

Pour répondre à la première question nous avons installé sur nos machines l'outil JLink permettant de faire détecter la carte, ainsi que OpenOCD servant au flashage de celle-ci. Il convient de préciser que l'installation de ces deux outils a été assez fastidieuse et a nécessité plusieurs séances de travail.

Pour la deuxième interrogation, nous avons opté pour Cpputest , ses possibilités de mock et test unitaires nous paraissant des plus pertinentes pour notre projet.

Au début du projet, nous ignorions que nous allions devoir travailler avec NuttX, et nous avons donc orienté nos recherches pour programmer directement la carte , ainsi chacun des trinômes a effectué des recherches propres concernant la mission qu'il lui était dédiée.

3 LE SERVOMOTEUR

Le premier trinôme chargé de mettre en place le servomoteur a donc effectué des recherches approfondies dans la documentation de la SAM4S, à commencer par le fonctionnement général de la carte, puis sur les domaines approfondis tels que la génération de signal (PWM) ou encore les registres propres à la carte.

Suite à ces recherches, nous avons mieux appréhendé le comportement de PWM et nous avons pu également voir les fonctions importantes qui nous seraient utiles dans la suite du projet.

Sur ce point, on peut citer par exemple « start », « ioctl », « open », nécessaires à l'utilisation de PWM. Cette recherche nous a également permis de localiser la PIN du générateur de signal.

Il est également important de spécifier que des recherches sur le servomoteur en lui-même ont été effectuées par ce trinôme.

4 LE CAPTEUR DE LUMIÈRE

Pendant ce temps, le second trinôme qui avait la tâche de mettre en place la mesure de la lumière, a commencé par rechercher la documentation relative au « TEMT 6000 », le capteur de lumière que l'équipe pédagogique avait acheté pour l'occasion .

Après analyse de la documentation sur celui-ci, le groupe est arrivé à la conclusion que le signal que nous allions récupérer du capteur était un signal de type « analogique ». Une fois cette information acquise, il s'est imposé que devions effectuer des recherches sur le « adc » (Analog to digital converter), soit, en français , le convertisseur analogique numérique.

Il s'est heureusement trouvé que la SAM4S possède un circuit qui effectue la conversion analogique numérique, de cette façon nous n'aurons donc pas besoin de mettre en place un convertisseur externe

Après une rapide recherche dans le manuel de la SAM4S, nous avons été en mesure de localiser la PIN du convertisseur sur la carte (Block J2) .

Une fois la PIN localisée, il fallait être en mesure de récupérer la valeur sur celle-ci ; et nous avons donc recherché un exemple de programme capable de lire et faire fonctionner l'ADC correctement.

Après quelques recherches nous avons trouvé le programme suivant : <ADC Sam4S>

C'est par la suite, quand nous avons appris que nous allions utiliser Nuttx, que le travail s'est complexifié . En effet, nous avons dû compiler NuttX avec une configuration propre pour la carte, ce qui n'a pas été évident dans la mesure où nous ne l'avions jamais utilisé auparavant. Après de nombreux essais infructueux, nous sommes enfin parvenus à « builder » celui-ci avec le bon « KConfig ».

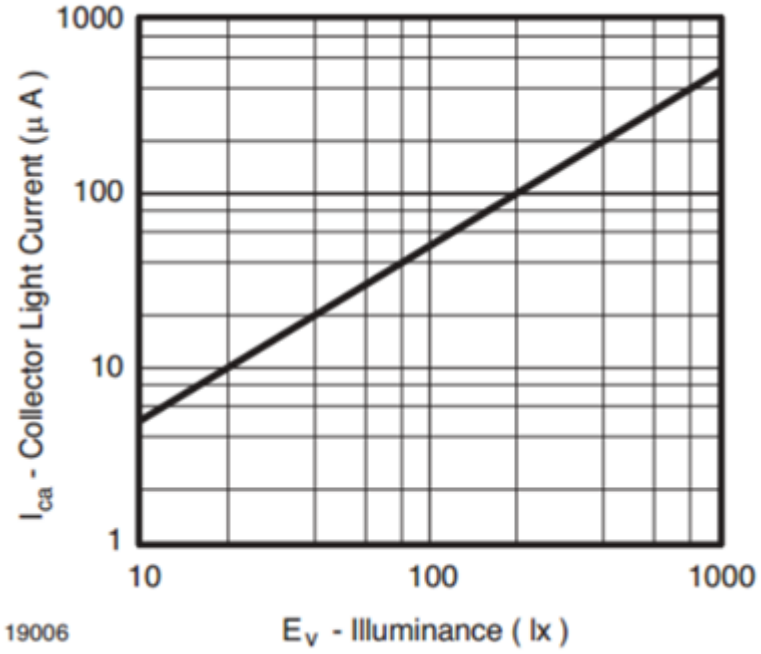
Une fois notre Nuttx compilé et flashé , nous y avions accès via un terminal putty ; et nous sommes mis en quête d'un exemple d'utilisation de l'adc pour nuttx.

Nos recherches nous ont menés à l'exemple suivant :
<ADC Nuttx>

L'analyse de celui-ci nous a demandé un certain temps d'étude.

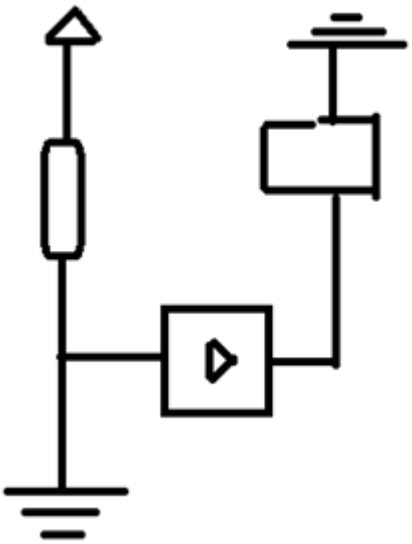
Pour finir , nous avons appris qu'il serait impossible de faire marcher l'adc dessus faute de drivers, et nous avons donc décidé de faire abstraction de la carte et tenté de « mocker » celle-ci.

Pour « mocker » efficacement le capteur et la carte, il s'est avéré que nous avions impérativement besoin d'informations concernant ces deux éléments :



Il s'agit ici de la courbe du signal en sortie du capteur.

L'intensité varie donc en fonction du nombre de lux en entrée du capteur. Le convertisseur adc de la carte devant , lui, mesurer une tension, nous avons fait la liaison entre ces deux mesures physiques avec une résistance de tirage .



Calcul de la résistance de tirage : La tension maximale en entrée du convertisseur adc est de 3.6V, nous avons donc effectué le calcul de la résistance de tirage de la manière suivante :

$U = R * I$
$3.5 = R * 0.001$
$(3.5/0.001) = 3.5\text{kOhms}$

On a donc une résistance de tirage de 3.5kohms Une fois ces informations acquises, nous étions en mesure de « mocker » la carte et le capteur :

5 LE CAPTEUR DE COLORIMÉTRIE

Un fois ce capteur terminé, nous avons décidé de passer sur le capteur colorimétrie , étant précisé que nous avions la tâche de le choisir par nous-même. Nous avons choisi S9706, il s'agit d'un capteur numérique 12 bit ,des nuances de rouge de vert et de bleu sur 4 bits chacune , et au moment de ce choix, nous avons bien vérifié que la tension de celui-ci était nettement inférieure à celle de la carte. A partir de ces informations nous avons effectué les test et les mocks associés à la carte.

6 SUITE SERVOMOTEUR

Une fois le pin localisé, il nous fallait des données spécifiques susceptible de fonctionner avec notre servomoteur.

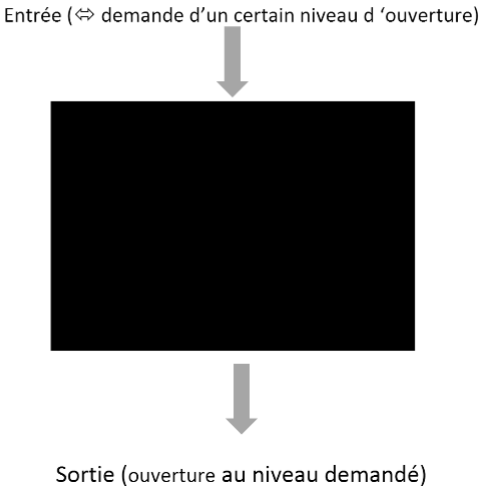
Dans cette perspective, nous avons récupéré les données telles que la fréquence (20 ms) ou bien le voltage etc ... Étant donné notre changement de trajectoire causé par NuttX, il était temps de se lancer dans les « mocks ». Le système de fonctionnement des « mocks » est le fait de « bouchonner » les fonctions que l'on ne peut pas utiliser ; dans notre cas NuttX n'étant pas fonctionnel, nous avons donc dû « bouchonner » les fonctions NuttX.

Nuttx est doté de plusieurs fichiers et exemples qui permettent de comprendre son fonctionnement global. Nuttx a pour rôle d'être une surcouche gérant les registres-systèmes, soit le plus bas niveau de l'application. Dans cette optique, nous avons épluché les .h et .c afin de mieux comprendre les fonctionnements du pwm.

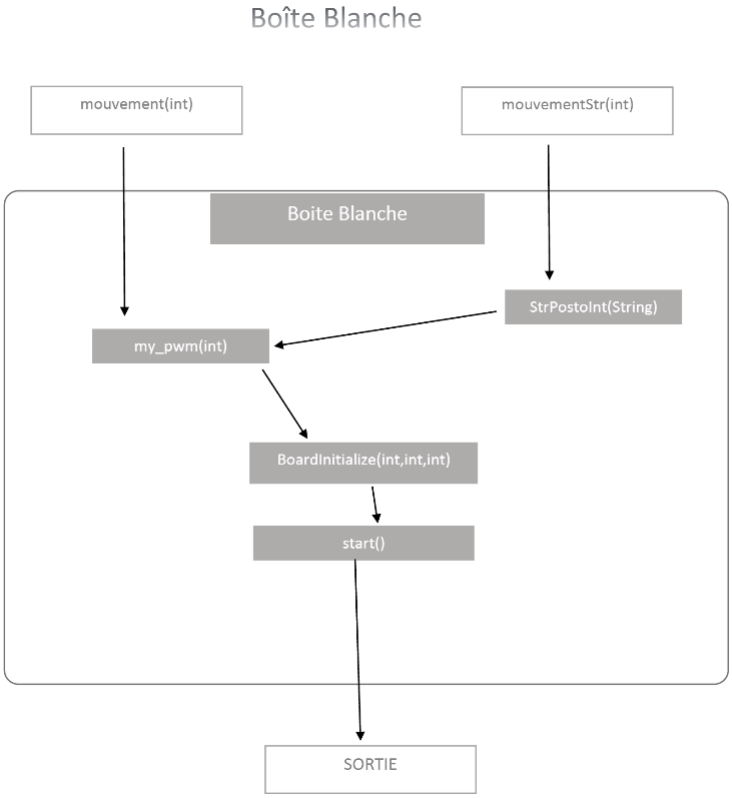
En premier lieu nous avons localisé les fonctions essentielles à nos besoins, comme « start(),open() », dans le but de les mocker. Dans l'ordre, nous avons établi la boîte noire, étant donné que toutes les fonctions étaient trop liées au matériel, nous ne pouvions pas les coder directement. La boîte noire, c'est-à-dire le système très sous forme schématique, nous a permis de nous donner une

bonne vue d'ensemble, et de savoir de plus en plus ou on allait.

Schéma de la Boîte Noire



Nous avons donc réalisé et fait fonctionner les « mocks » du côté boîte noire. Une fois la boîte noire terminée, il nous a fallu passer du côté boîte blanche, qui est le système détaillé ; c'est-à-dire, l'enchaînement de processus avec chaque sous « modules », contrairement à la boîte noire ou l'on ne dispose que des éléments les plus importants.



Nous avons en parallèle établi les « mocks » côté boîte blanche et en même temps, réalisé le code fonctionnel. Bien entendu nous avons réalisé des tests pour nous assurer du bon comportement des fonctions utilisées.

7 GESTION DE PROJET

Pendant ce projet, nous avons travaillé avec un gestionnaire de version et un outil d'intégration continue. Pour finaliser le choix de ce dernier, nous avons discuté des avantages et des inconvénients en groupe, et avons finalement opté pour jenkins.

NB :des captures d'écran de Jenkins sont disponible dans le GitHub(Doc)

En effet, celui-ci était sensé nous permettre de faire des tests en réel sur la carte ; mais au final nous n'avons travaillé qu'avec des « mocks », et nous aurions donc pu nous contenter de travis.

Il est à préciser que nous ne nous sommes pas limités à utiliser les différents outils de gestion , mais nous avons également effectué des réunions régulières, permettant de prévoir des plannings .

OBJECTIFS DU VENDREDI 3 AVRIL APRÈS-MIDI

- Demander pour le type de tests à produire , Simon, Rémy
- Produire des tests en conséquence , Simon, Rémy

OBJECTIFS DU JEUDI 2 AVRIL 2015 - MATIN

- Demander concernant l'avancée avec les drivers – Thibaud Viard
- Choisir Jenkins / Travis – Groupe
- Mettre en place le support choisi – Mickaël
- Réaliser des mock et tests unitaires – Simon , Rémy
- Écrire le programme pour récupérer la tension ADC (bits) – Groupe Luminescence
- Écrire le programme gérant le pwm – Imade, Raphaël
- Vérifier validité des tests (valeur de retour / entrée)

8 CONCLUSION

En conclusion, bien que nous n'ayons pas réussi à atteindre notre objectif initial, c'est-à-dire de tout faire fonctionner sur le carte, la réalisation de ce projet en équipe s'est avérée très pédagogique pour chacun d'entre nous, tant sur le plan de l'électronique que sur celui de l'informatique.

L'idée de partir d'un projet virtuel pour arriver à une application concrète est également extrêmement motivante. Un autre effet positif de ce projet réside en la valorisation du travail en groupe, chacun devant gérer ses propres tâches et dialoguer avec les autres pour opérer les meilleurs arbitrages pour le bien commun. Certains se sont découverts une stimulation positive

au travail, d'autres ont tiré des leçons d'humilité et fait un certain apprentissage de la démocratie (à une échelle modeste. . .).

Même si les débuts de la conception du projet ont été ardus et chronophages pour chaque difficulté d'ordre mathématique ou informatique, nous avons su chercher, tester et appliquer les meilleurs outils, afin d'aboutir aux solutions idoines. Il est intéressant de préciser également, que certains "blocages" informatiques nous ont obligés à remettre en cause notre manière de penser certains algorithmes, et ont ébranlé quelques-unes de nos certitudes.

Le travail en équipe a bien fonctionné, et cela constitue une satisfaction à titre personnel pour chacun d'entre nous. Chaque individu s'est positionné dans le groupe à la place qui lui convenait, et ce, pour être le plus efficace, mais dans un esprit de parité. Les échanges ont été riches et constructifs, et la mutualisation des connaissances a parfaitement fonctionné.

Nous sommes conscients que dans nos futurs métiers, de nombreux travaux reposent sur l'esprit d'équipe et le travail de groupe, dans ce sens cette expérience positive est de bonne augure .

Bien que nous soyons globalement satisfaits du résultat final, il doit être précisé que si nous avions disposé de temps supplémentaire, nous aurions aimé pouvoir faire plus.