

SYS3046 Microcontrôleurs : périphériques et applications

C. TRABELSI & Y.SELLAMI

Projet

1 Objectif du projet

Ce projet consiste à concevoir un système complet, avec ses parties matérielle et logicielle, autour du microcontrôleur Cortex-M0+ de la carte STM32L053R8 **en binôme**. La conception commence par imaginer l'idée du projet, choisir les composants à utiliser et schématiser les connexions avec la carte microcontrôleur en utilisant le logiciel Kicad. Après la conception de l'architecture du système, il faut écrire la partie logicielle (les drivers des composants) et le main de l'application dans l'environnement STM32CubeIDE. Il est recommandé d'écrire la partie logicielle d'une façon progressive, en testant un composant à la fois. Pour test, il faut connecter le composant à la carte, lancer le débogage et vérifier que le programme réagit correctement avec le composant.

Durant les dernières séances de ce module (**semaines du 02 mai et du 09 mai**), chaque binôme présentera son travail avec une démo de 5min. Les comptes rendus finaux (dépôts sur Moodle) sont attendus pour le **15 mai** à 23h59 au plus tard.

2 Déroulement du projet

2.1 Première étape : Conception du projet

En binôme, concevez une idée de projet autour de la carte STM32L053R8 et en mettant en œuvre au minimum un capteur ou un actionneur de la liste ci-dessous :

- Les capteurs :
 - Capteur de distance infrarouge Sharp GP2Y0A21YK0F
 - Capteur de distance ultrason HC-SR04
 - Boutons poussoirs
 - Clavier matriciel ECO.16250.06

- Potentiomètre
- Photorésistance (capteur de luminosité) LDR1000
- Les actionneurs :
 - Moteur pas à pas 440-436 et son driver Pololu A4988
 - Servomoteur SG90 9g MicroServo
 - Afficheurs LCD (trois modèles disponibles : Displaytech 162, DEM 16216 et JM162A)
 - Afficheur 7-segments anode commune + décodeur inverseur SN7447A
 - LEDs
 - LEDs RGB L-154A4SURKQBDZGW
 - Buzzer KPEG200A

La plupart de ces composants ont déjà été présentés dans le cours. Leur fonctionnement est présenté dans le cours 3-4-5. Pour avoir plus de détails techniques sur ces composants, vous avez la documentation disponible dans la section « Docs des composants » sur Moodle.

Exemples de fonctionnalités qui peuvent vous inspirer

- Moteur pas à pas : faire avancer le moteur d'un certain nombre de pas, alterner les modes pas et demi-pas, varier la vitesse du moteur, tourner dans les deux sens, etc.
- Servomoteur : faire tourner le moteur d'un angle, tourner dans les deux sens, etc.
- Capteur infrarouge/ultrason : détecter la présence d'un obstacle ou d'un objet, déterminer la distance qui sépare votre système d'un objet, faire varier la luminosité d'une LED ou la vitesse d'un moteur en fonction de la distance qui sépare votre main du système, etc.
- Afficheur 7-segments : afficher une distance à partir d'un capteur infrarouge/ultrason, etc.
- LEDs : implémenter un simulateur de feux tricolores avec un bouton piéton, utiliser des LEDs pour indiquer dans quel mode est le système, varier la luminosité de la LEDs, etc.

Après vous être mis d'accord sur l'idée du projet, vous devez valider l'idée par l'encadrant et lui communiquer la liste de composants que vous voulez. Une fois

l'idée validée, vous pouvez récupérer les composants auprès du technicien qui va noter lui aussi ce que vous avez récupéré.

2.2 Deuxième étape : Schématisation de la partie matérielle du projet

Après avoir fixé l'idée du projet, schématisez l'architecture de votre système dans le logiciel Kicad. Pour cela, consultez le document « Guide de démarrage Kicad » dans le dossier « Schématisation sur Kicad » de la section « Projet » sur Moodle.

Une fois vous avez schématisé l'architecture matérielle de votre projet sur Kicad, il faut appeler l'encadrant pour validation.

2.3 Troisième étape : Développement de la partie logicielle du projet et test sur l'architecture physique

Partez du projet simpliste créé sur STM32CubeIDE dans les TPs. Pour chaque composant matériel de votre projet, écrivez un pilote (un fichier .h et un fichier .c) qui implémente les fonctions d'initialisation et les différentes opérations effectuées par ces composants. Pour implémenter les fonctions d'initialisation, inspirez-vous de la fonction « Led_init » du fichier led.h vue en TP pour la configuration des pins en sortie et la fonction « Button_init » pour les pins configurées en entrée.

La plupart des composants de la liste ont déjà été présentés dans le cours. Leur fonctionnement est présenté dans le cours 3-4-5. Pour avoir plus de détails techniques sur ces composants, vous avez la documentation disponible dans la section « Docs des composants » sur Moodle.

La partie logicielle de votre projet doit être la plus modulaire possible. Pour cette raison, privilégiez l'implémentation des fonctionnalités qui impliquent les différents composants sous forme de fonctions/procédures à écrire soit dans le fichier main.c soit dans des fichiers .h et .c séparés (les fonctionnalités impliquant un seul composant sont à écrire dans le pilote de ce composant).

Il est recommandé d'écrire la partie logicielle d'une façon progressive, en testant un composant à la fois. Pour test, il faut connecter le composant à la carte, lancer le débogage et vérifier que le programme réagit correctement avec le composant.

Si votre composant ne donne pas le résultat souhaité, vérifiez deux éléments :

- Votre programme donne les bonnes valeurs des signaux en vérifiant les registres à l'aide du débogueur dans la fenêtre SFRs
- Votre composant interagit correctement en regardant à l'aide de l'oscilloscope les

entrées/sorties du composant.

3 Rendu du projet

A la fin du projet, vous devez rendre un dossier qui contient :

- un rapport en **pdf**
- les codes source de votre projet.
- Un lien vers une vidéo de démonstration

Ce dossier est à déposer pour le **15 mai à 23h59** au plus tard.

Le rapport doit contenir :

- a) Une page de garde
- b) Un sommaire
- c) Une introduction
- d) Le projet :
 - Présentation de l'objectif du projet et de ces principales fonctionnalités
 - Organisation et répartition des tâches
 - L'architecture matérielle schématisée sur Kicad et sa description.
 - Description de la structure de votre programme et la façon dont vous avez implémenté les différentes fonctionnalités
 - Illustration des fonctionnalités principales à l'aide de photos ou/et des captures d'écran
- e) Une conclusion :
 - Décrire brièvement les connaissances et apprentissages acquis pendant cet exercice.
 - Énoncer également les difficultés principales surmontées pour arriver au résultat souhaité.
 - Présenter les améliorations possibles du projet