INTRODUCTION

On trouve de nos jours des robots dans tous les domaines : industrie, santé, transport et même dans la sécurité, un secteur dont se préoccupent de façon croissante les entreprises et les particuliers. Leur développement fait appel à plusieurs métiers de l’électronique et constitue donc un cadre idéal pour l’évaluation d’étudiants issus de filiales complémentaires en vue d’un recrutement.

C’est dans ce contexte qu’est proposé ce stage de fin d’études : Elsys Design a décidé de mobiliser une équipe pluridisciplinaire autour d’un projet dont le but est de développer un robot autonome de surveillance.

Je vais vous présenter aujourd’hui toute la partie de conception FPGA que j’ai réalisé lors de ce stage

PRESENTATION DU CONTEXTE

J’ai réalisé mes 6 mois de stage chez Elsys Design qui est une société d’ingénierie spécialisée en conception des systèmes électroniques. Elle rassemble près de 800 ingénieurs au sein de 11 implantations situées majoritairement en France, mais aussi en Serbie et dans la Silicon Valley.

C’est une branche d’Advans group qui réunit plusieurs autres sociétés expertes dans les domaines du logiciel et de la mécanique.

Son activité se décompose en trois parties :

* L’activité Forfait (développements dans les bureaux d’étude d’ELSYS Design pour des clients extérieurs)
* L’activité Assistance Technique (les ingénieurs ELSYS Design renforcent les équipes des clients dans le cadre de missions régie)
* L’activité R&D et innovation

ELSYS Design possède un large spectre de compétences dans tous les métiers du logiciel et de l’électronique, dont les systèmes embarqués, la microélectronique et la carte électronique.

Dans le but de développer un robot de surveillance autonome, une équipe de trois stagiaires a été mobilisée :

* Un concepteur Hardware : responsable du développement d’une carte interface, permettant de connecter les différents composants, et du développement d’une station de charge.
* Un développeur Software : responsable de la programmation logicielle du robot et de l’IHM
* Une conceptrice FPGA : responsable de la programmation de la partie PL (Programmable Logic) du SoC. Il définit l’architecture firmware (architecture, conception, code, test, vérification et routage).

Elsys design s’est placé dans le rôle du client, nous avons dû faire l’intégralité du projet en cycle en V : il nous fallait donc rédiger des spécifications et un planning, gérer un budget ainsi que de mettre en place des réunions régulièrement pour les encadrants en plus de toute la partie technique.

Il nous a fallu partir de la base holonome montrée ici. Base développée en 2019 par une étudiante en alternance .

SPECIFICATIONS

Il nous a fallu réfléchir à toute la partie spécifications, notamment le choix des composants pour pouvoir se localiser, éviter les obstacles… Nous avons fait les choix suivants :

* L’IMU, l’odométrie déjà présent dans la base holonome du robot et le lecteur RFID : les trois ont été utilisés de manière complémentaire.
* Les ultrasons pour l’évitement d’obstacles
* Des ESP 32 et des capteurs de mouvement pour la détection d’intrusion

Le choix de la carte a été la Zybo Z7 20 car elle nous permettait d’avoir de nombreuses GPIOs, mais elle avait aussi une connectique assez complète ce qui nous aurais permis de pouvoir rajouter de nouveaux périphériques si le temps l’avait permis. Notamment une caméra, des enceintes, microphones…

Logiciels utilisés…

ARCHITECTURE FONCTIONNELLE

Une fois les spécifications fixées il a été important de réfléchir au partage des tâches, notamment entre le software et le FPGA. L’idée initiale de partir sur la caméra a été abandonnée car cette partie aurait pu être très chronophage sur FPGA,

DIJKSTRA

RFID

Connexion : Le lecteur émet un champ électromagnétique, lorsque le récepteur passe dans ce champ, l’énergie captée lui permet de s’alimenter et de répondre aux requêtes de l’émetteur (lecteur).

Activation : Cette étape est d’ordre logiciel et est donc couvert par les couches 3 et éventuellement 4 de la norme.

L’émetteur envoi une requête d’identification appelée “REQ” et attends une réponse appelée “ATQ”. Comme nous l’avons vu plus haut, il existe deux variantes de la norme A et B. L’émetteur ne pouvant pas identifier la variante à utiliser, il procède successivement à une requête REQA puis REQB.

Si le récepteur répond à la requête REQA avec une réponse ATQA, alors c’est une variante A, s’il répond à la requête REQB avec une réponse ATQB, alors c’est une variante B.

Une fois l’identification de la variante faite, l’émetteur doit identifier le type de récepteur et activer la carte.

Il procède à la boucle d’anticollision (Anticollision Loop). Cette procédure permet en fait de lier l’émetteur à un seul récepteur. C’est important puisqu’il est possible d’avoir plusieurs récepteurs dans le champ magnétique de l’émetteur.

* UID : unique ID
* SAK : select acknowledge