

TD « périphérique de μC », application voilier

Objectifs :

- câbler les IO du μC sur les capteurs/actionneurs du voilier,
- concevoir le logiciel embarqué répondant au cahier des charges (cad concevoir l'architecture logicielle et définir les fonctionnalités de chaque bloc logiciel)
- répartir les diverses activités au sein de chaque équipe

Prérequis :

- avoir une bonne maîtrise des périphériques fondamentaux (GPIO, Timer, interruptions, PWM),
- avoir lu et compris le *Dossier technique voilier 2.0* (*DossierTechVoilier2_0.pdf*),
- avoir lu et compris le *cahier des charges du projet* (*CahierDesChargesVoilier2020.pdf*),

1. Synthèse du cahier des charges

<i>Ref Fonctionnalité</i>	<i>Nom fonctionnalité</i>
<i>F1</i>	Bordage automatique
<i>F2</i>	Orientation Voilier, cap
<i>F3</i>	Système anti-chavirement
<i>F4</i>	Transmission d'informations

2. Les étapes conception

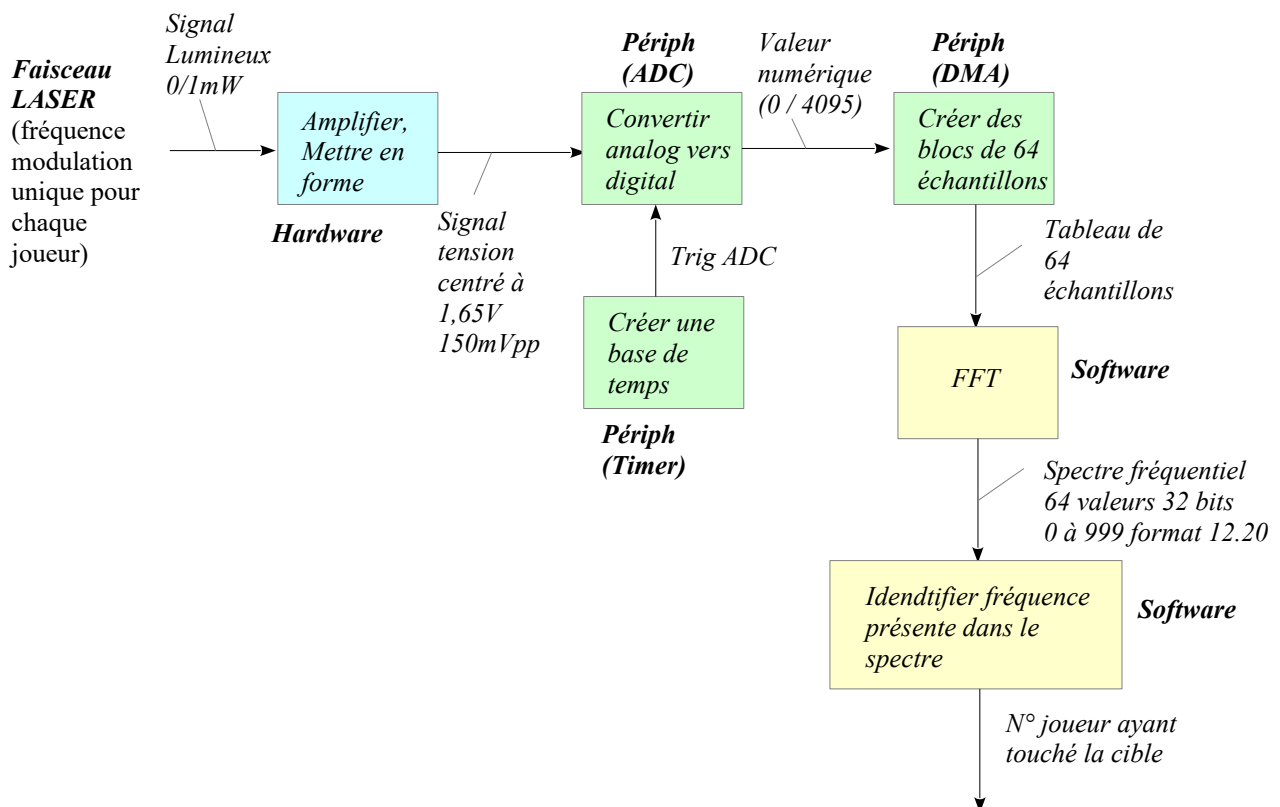
2.1. Elaboration de schéma bloc hard/soft pour chaque fonctionnalité

Pour chacune des fonctionnalités,

- établir un schéma fonctionnel (bloc hardware et software) qui permet la réalisation de la fonctionnalité,
- Donner les min max des grandeurs d'entrée / sortie ainsi que leur nature.

On donne un exemple :

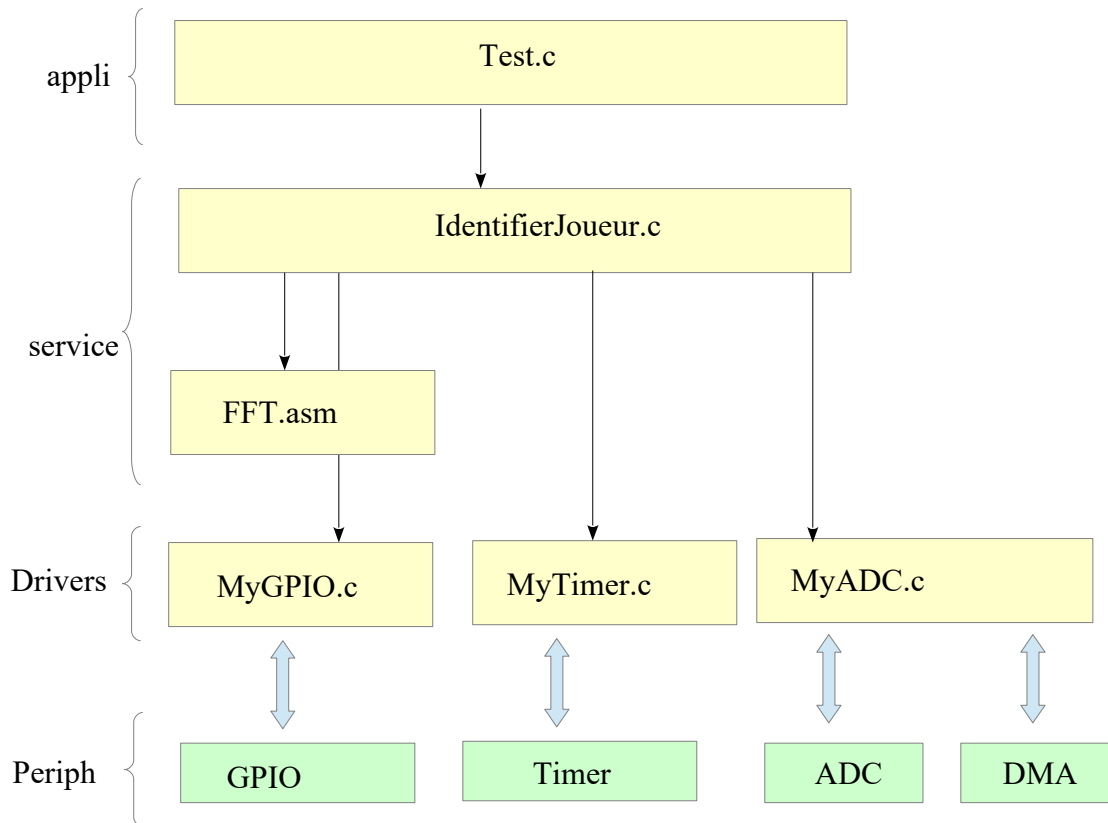
Fonctionnalité : Identifier le joueur qui a touché la cible (contexte d'un jeu type laserquest)



2.2. Elaborer l'architecture logicielle de la fonctionnalité

Exemple :

Architecture logicielle de la fonctionnalité : *Identifier le joueur qui a touché la cible*



2.3. Définir le périphérique précis utilisé, ainsi que les IO choisis

→ définir également la configuration précise des GPIO

Exemple :

ADC choisi : *ADC1*

Timer choisi : *TIM3*

GPIO choisi : *PA1* configuré en analog input

Peut être fait sous la forme d'un tableau.

3. Synthèse ressources matérielles

Etablir un même tableau de synthèse qui, pour chaque fonctionnalité donne la liste des périphériques utilisés et des *IO* utilisées. L'objectif de cette synthèse est de bien vérifier si les ressources sont bien partagées, qu'il n'y a pas de conflit ni dans les périphériques, ni dans les *IO*.

4. Synthèse logicielle

Réunir sur un même “diagramme de classes” l'ensemble des éléments logiciels nécessaires pour réaliser l'ensemble des fonctionnalités.

Expliciter de manière précise la fonction, le rôle de chaque bloc logiciel. Cela peut se faire sous la forme d'un tableau.

5. Synthèse temps réel

Proposer un séquençement dans le temps qui montre l'enchaînement de toutes les tâches logicielles, et donc la réalisation dans le temps des diverses fonctionnalités.

C'est à ce niveau que devraient être définies les priorités des diverses interruptions qui vont intervenir.

6. Répartition des activités

Etablir une répartition des tâches en matière de réalisations logicielles. Prévoir dès le départ la manière dont chaque bloc va être testé.

Là encore, un tableau peut aider à la lisibilité.