

# M2 EEA SME

Projet Barre Franche - Novembre 2019

---

## EIEAS3GM SYNTHESE ET MISE EN ŒUVRE DES SYSTEMES

---

**Auteurs :**

Nicolas OTAL

Antoine ROUTIER

**Encadrants :**

M. PERISSE



UNIVERSITÉ  
TOULOUSE III  
PAUL SABATIER



Université  
de Toulouse

## Introduction

*Dans le cadre de notre UE "Synthèse et Mise en œuvre de système", un bureau d'étude nous a été proposé avec pour objectif de mettre en œuvre une solution logicielle/matérielle qui répond au besoin de gestion et contrôle de trajectoire d'un voilier de barre franche.*

*A l'aide des bases acquises, durant notre formation, en VHDL, simulation, et conception systèmes l'objectif de cette UE sera la réalisation d'un contrôleur de Barre-France de voilier par FPGA (Altera). Il sera nécessaire d'étudier, décomposer, coder et implémenter chaque fonctions une à une et d'intégrer la globalité du projet à l'aide d'un Bus Avalon permettant l'interconnexion des fonctions au MCU intégré au FPGA.*

*Pour réaliser cela, nous avons respecter le processus de développement consistant à réaliser une analyse des besoins et du contexte permettant d'identifier les différentes interfaces du système, nous avons par la suite réaliser la conception du système en décomposant notre fonction principale en différents blocs. Ces différents blocs ont fait l'objet d'une description fonctionnelle avant d'être implémenter. Suite à cela, nous avons réaliser un ensemble de simulation et test sur maquette pour vérifier le bon fonctionnement de chaque module intégré pour valider finalement notre projet sur une maquette.*

## Sigles et acronymes

<b>FPGA</b>	<i>Field Programmable Gate Arrays</i>
<b>GPIO</b>	<i>General Purpose Input/Output</i>
<b>GPS</b>	<i>Global Positioning System</i>
<b>NMEA</b>	<i>National Marine Electronics Association</i>
<b>SOPC</b>	<i>System On Programmable Chip</i>
<b>SRAM</b>	<i>Static Random Access Memory</i>
<b>UART</b>	<i>Universal Asynchronous Receiver Transmitter</i>
<b>VHDL</b>	<i>VHSIC Hardware Description Language</i>
<b>VHSIC</b>	<i>Very High Speed Integrated Circuits</i>

## Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Cahier des charges du Projet "Barre-Franche"</b>	<b>5</b>
1.1 Introduction technique . . . . .	5
1.2 Cahier des charges général . . . . .	5
1.3 Cahier des charges technique . . . . .	6
<b>2 Conception Matérielle</b>	<b>7</b>
2.1 Réalisation fonction simple - Gestion anémomètre . . . . .	7
2.2 Réalisation fonction complexe - Gestion vérin . . . . .	9
2.3 Réalisation interfaces Avalon . . . . .	9
<b>3 Conception conjointe Matérielle/Logicielle</b>	<b>10</b>
<b>4 Développement Code Embarqué</b>	<b>12</b>
<b>5 Validation des Fonctions</b>	<b>13</b>
<b>6 Conclusion</b>	<b>14</b>

# 1 Cahier des charges du Projet "Barre-Franche"

## 1.1 Introduction technique

Le projet qui nous est demandé est basé sur les SoC et plus particulièrement sur un FPGA, de chez Altera, embarqué dans un voilier pour en piloter la barre-franche. Ce dispositif électronique fonctionnera à l'aide de différentes entrées/sorties (gyroscope, anémomètre, GPS, convertisseur analogique/numérique, vérin, boutons, buzzer).

## 1.2 Cahier des charges général

Durant ce projet, nous allons utiliser les différentes compétences acquises à travers les différents cours de l'année et les mettre en corrélation pour mener à bien ce dernier. Le projet devra respecter certains critères présentés ci-dessous :

1. Le dispositif devra utiliser un appareil de mesure pour capter la valeur de la vitesse du vent.
2. Le dispositif devra utiliser un appareil de mesure pour capter la direction du vent.
3. Le dispositif devra réceptionner des données GPS, traiter ces données et agir sur le dispositif en fonction des résultats obtenus après traitement.
4. Le dispositif devra avoir une interface entre l'opérateur et le voilier (boutons, Leds et buzzer).
5. Le dispositif devra piloter la barre-franche du voilier à l'aide des différents appareils pour venir piloter un vérin (Vérin et compas).

La figure 1 modélise le système qui sera réalisé avec ses différentes fonctions et les différents flux d'informations nécessaires au système "Barre-Franche".

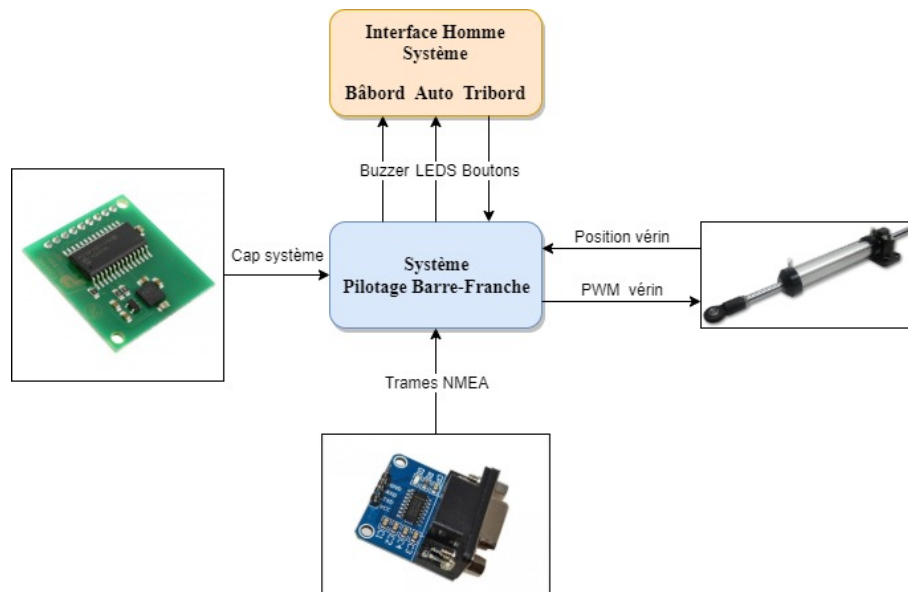


FIGURE 1 – Diagramme contexte du système

Pour des raisons pratiques et de temps nous avons implémenté la mesure de la vitesse du vent, la gestion du vérin, la gestion du compas ainsi que l'asservissement du vérin, nous n'avons pas implémenté la gestion des trames NMEA, mais avons ajouté l'interface hommes système (boutons, leds et buzzer).

### 1.3 Cahier des charges technique

Le projet se porte sur la réalisation d'un dispositif embarqué à base de FPGA, de chez Altera. Il sera composé de deux parties principales une partie Hardware sur le FPGA et une partie Software intégrée/développée dans le FPGA (SOPC). Les deux parties communiqueront par le biais du Bus Avalon, qui est le Bus développé par Altera pour leur SOC. Le cœur du projet sera composé d'un FPGA Cyclone IV EP4CE22F17C6N du fondeur Altera permettant le développement du projet Hardware et Software sur la même carte d'évaluation. Différentes fonctions seront implémentées :

1. Une fonction qui permettra de lire la mesure de la vitesse du vent (0-250km/h). La fonction devra lire la sortie de l'anémomètre qui est une sortie logique de fréquence variable (0 à 250 Hz).
2. Une fonction générant un signal PWM qui sera utilisé dans plusieurs parties du projet. Cette fonction sera intégré plus tard dans le SOPC du FPGA permettant la génération d'un signal PWM qu'on utilisera au travers du Bus Avalon.
3. Une fonction de gestion vérin gérant le pilotage de la barre franche.
4. Une fonction qui utilise un compas pour récupérer les mesures d'angles sur le plan horizontale du voilier, permettant de donner un cap à celui-ci.
5. Une fonction qui permet la gestion de l'interface Homme système (composée de différents boutons[Bâbord, Tribord, Auto/Manuel], des LEDS ainsi qu'un buzzer).
6. Une implémentation d'un MCU dans le FPGA grâce à l'outil SOPC du logiciel d'Altera. Celui-ci permettra le traitement et l'affichage des différentes variables du projet (cap du voilier, position vérin, position GPS, gestion des PWM et la vitesse du vent).

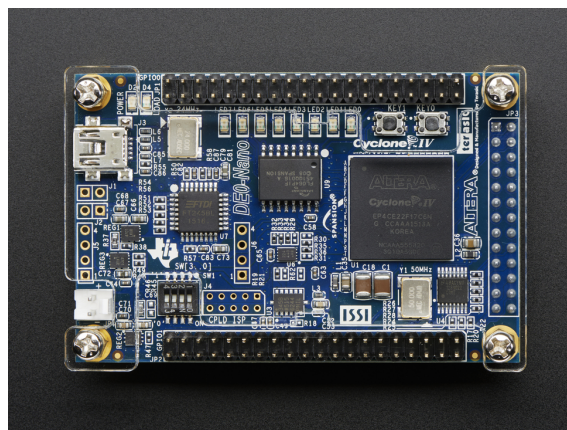


FIGURE 2 – DE0 Nano Altera

## 2 Conception Matérielle

Pour mettre en oeuvre les différentes parties du projet combinatoires et séquentielles, nous avons du passer par des étapes d'analyses comportementales de chaque blocs à réaliser pour ainsi créer des composants simples pouvant être utiliser à la composition de fonctions plus complexes plus tard. L'implémentation de chaque composants a été réalisée grâce au langage VHDL sur le logiciel de développement Quartus.

Ainsi nous avons réalisé différentes "boîtes" de composants simples pour réaliser des fonctions entières, et plus complexes, contenant les blocs simples.

### 2.1 Réalisation fonction simple - Gestion anémomètre

La fonction réalisant l'acquisition de la vitesse du vent (0 à 250 Km/h) se fait à l'aide d'un anémomètre qui sert de transducteur convertissant la vitesse du vent en fréquence variable (0 à 250 Hz).

Le composant doit fonctionner en deux modes :

1. Mode continu, dans ce mode le système actualise l'acquisition toutes les secondes de la vitesse du vent. Pour cela l'entrée "Continu" doit être à 1.
2. Mode mono-coup, dans ce mode le système effectue qu'une seule acquisition lorsque "Start/Stop" est activé. Une fois "data\_valide" envoyé le système remet les entrées "Start/Stop" et "data\_valide" à zéro après une acquisition et attend la prochaine entrée "Start/Stop".

La figure 3 représente la décomposition des composants utilisés pour la réalisation du bloc "Captage direction vent".

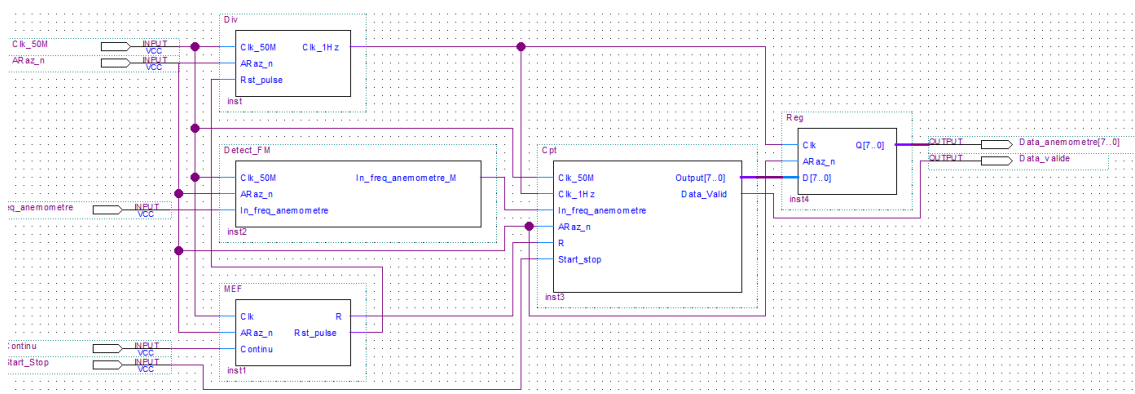


FIGURE 3 – Diagramme fonctionnel du circuit "Captation vitesse vent"

Le composant "Div" fournit une horloge de 1 Hz à partir de l'horloge 50 MHz donnée par la clock du FPGA. L'horloge de 1 Hz sera utilisée pour l'acquisition de la fréquence de l'anémomètre toutes les secondes.

Le deuxième composant "Detect\_FM" est en charge de détecter les fronts montants sortant de l'anémomètre pour transformer la fréquence générée en Hertz en une donnée interprétable par le composant "Cpt".

Le composant "Cpt" va permettre de récupérer la quantité de fronts montants comptés par le composant "Detect\_FM", toutes les secondes quand le composant sera en mode continue et une seule fois lors du mode mono-coup.

Le composant "Reg" permettant de stocker la valeur de la fréquence convertie dans un registre ce qui va permettre de venir traiter ou effectuer des calculs une fois celle-ci acquise (ici dans le MCU qui sera réalisé prochainement).

Le composant "MEF" est chargé de gérer le mode de fonctionnement du circuit (METTRE MEF)



## 2.2 Réalisation fonction complexe - Gestion vérin

Cette circuit est composé de 3 fonctions principales qui vont permettre le pilotage du vérin qui contrôle la barre franche du voilier :

1. Une gestion d'un signal PWM effectué par le process "PWM".
2. Un contrôle des butées réalisé par le process "Gestion\_Butée".
3. Une machine à états pour la gestion du convertisseur MCP3201.

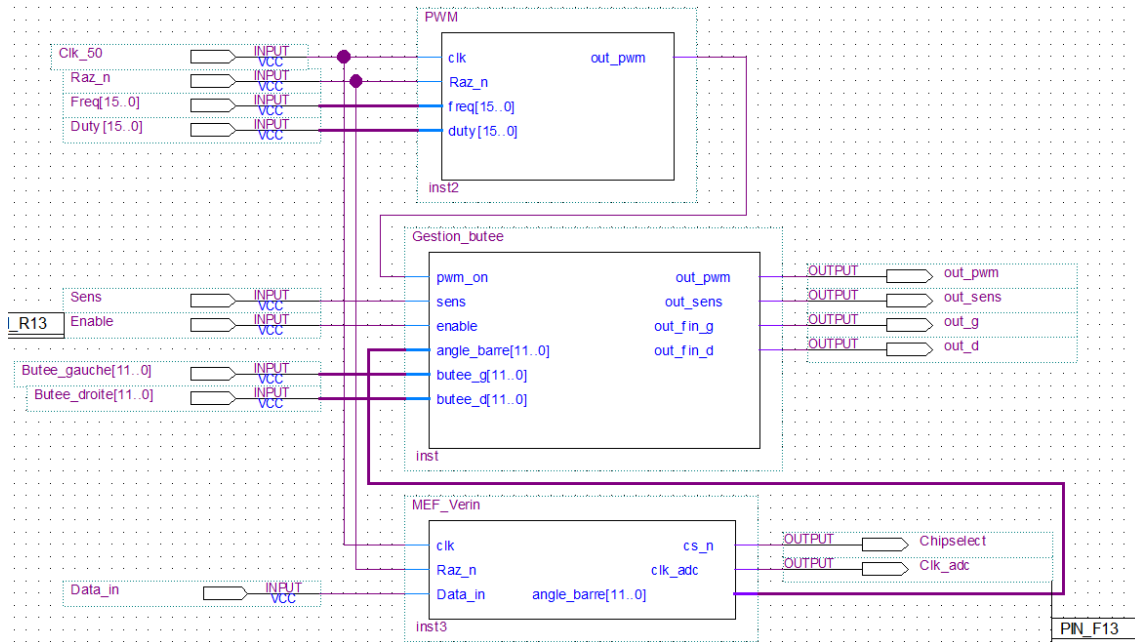


FIGURE 4 – Diagramme fonctionnel du circuit "Actionnement Barre"

Le premier composant "PWM".

Le composant "Gestion\_Butée".

Le composant "MEF" est chargé de gérer le mode de fonctionnement du circuit (METTRE MEF)

## 2.3 Réalisation interfaces Avalon

### 3 Conception conjointe Matérielle/Logicielle

La conception conjointe, est aussi appelée "codesign" qui correspond à une fonctionnalité complexe qui allie une partie logique programmée (flexibilité) et une logique câblée (performances). Utilisée pour réaliser des systèmes complexes sur Silicium ou SOPC, dans notre cas (System On Programmable Chip) sur FPGA. Ce qui permet d'intégrer les périphériques ainsi que le processeur dans le FPGA, dans notre projet il s'agit de la réalisation d'un micro-contrôleur.

Dans le système réalisé pour le projet nous avons généré, à l'aide de "SOPC Builder", un système SOPC comportant l'ensemble des fonctions du projet à partir de leurs fichiers VHDL. Dans le système on y retrouve le processeur NIOS II sur 32 Bits qui va permettre l'exécution des fonctions codées dans la partie logicielle, de la mémoire (type SRAM) de 20 Ko, un JTAG qui va permettre de créer une interface entre l'environnement de programmation (NIOS II) et le SOPC. Et pour finir un composant "SYSID" qui va permettre l'identification du système qui va protéger le système de tous mauvais téléchargements de programmes correspondant pas à l'application.

Conn...	Name	Description	Clock	Base	End	IRQ	Tags
	cpu_0	Nios II Processor	[clk]				
	instruction_master	Avalon Memory Mapped Master	clk_0				
	data_master	Avalon Memory Mapped Master	[clk]				
	jtag_debug_module	Avalon Memory Mapped Slave	[clk]				
	onchip_memory2_0	On-Chip Memory (RAM or ROM)	[clk1]				
	s1	Avalon Memory Mapped Slave	clk_0	0x00008000	0x0000cfff		
	sysid	System ID Peripheral	[clk]				
	control_slave	Avalon Memory Mapped Slave	clk_0	0x00011048	0x0001104f		
	Bouton	PIO (Parallel IO)	[clk]				
	s1	Avalon Memory Mapped Slave	clk_0	0x00011020	0x0001102f		
	Leds	PIO (Parallel IO)	[clk]				
	s1	Avalon Memory Mapped Slave	clk_0	0x00011030	0x0001103f		
	Pwm_avalon_0	Pwm_avalon	[clock]				
	avalon_slave_0	Avalon Memory Mapped Slave	clk_0	0x00011050	0x00011057		
	AvalonAnemo_0	AvalonAnemo	[clock]				
	avalon_slave_0	Avalon Memory Mapped Slave	clk_0	0x00011058	0x0001105f		
	AvalonVerin_0	AvalonVerin	[clock]				
	avalon_slave_0	Avalon Memory Mapped Slave	clk_0	0x00011000	0x0001101f		
	Gestion_boutons_0	Gestion_boutons	[clock]				
	avalon_slave_0	Avalon Memory Mapped Slave	clk_0	0x00011060	0x00011067		
	Vhdl_compass_0	Vhdl_compass	[clock]				
	avalon_slave_0	Avalon Memory Mapped Slave	clk_0	0x00011068	0x0001106f		
	jtag_uart_0	JTAG UART	[clk]				
	avalon_jtag_slave	Avalon Memory Mapped Slave	clk_0	0x00011040	0x00011047		

FIGURE 5 – Les différents composants du SOPC

Une fois les différents composants ajoutés avec leurs différents fichiers VHDL, une boite globale du projet est créée par "SOPC Builder".

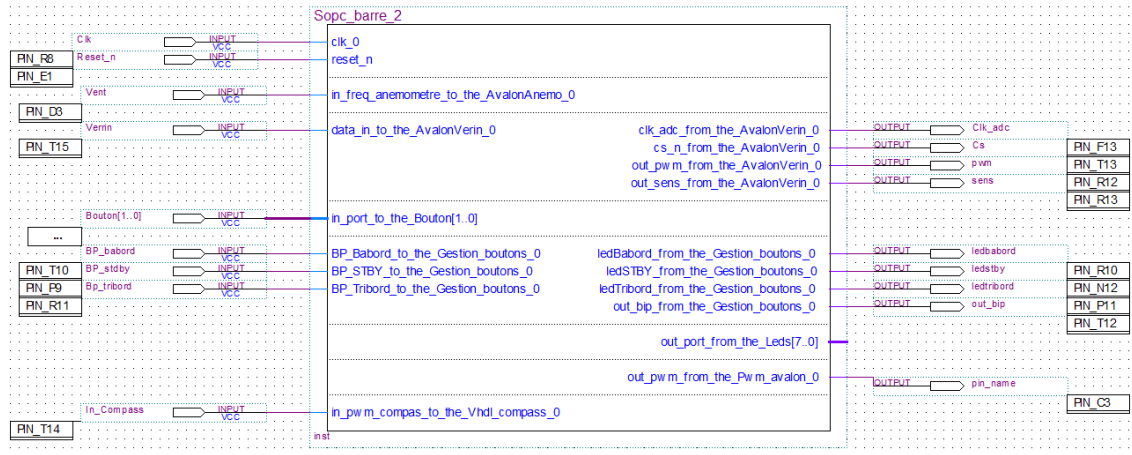


FIGURE 6 – SOPC et ses différents ports GPIO

## 4 Développement Code Embarqué

Dans cette section nous présenterons l'utilisation de NIOS II qui est l'outil de développement de Altera permettant la programmation, compilation et le debug du logiciel que nous implémenterons sur le micro-contrôleur de notre FPGA réalisé précédemment (sous environnement Eclipse).

## 5 Validation des Fonctions

Pour effectuer la validation de notre système nous avons téléversé notre projet sur le FPGA, la partie Hardware grâce au SOPC et la partie logicielle à l'aide de l'environnement Eclipse qui nous a permis de lancer/tester et déboguer notre projet directement sur la maquette d'évaluation.

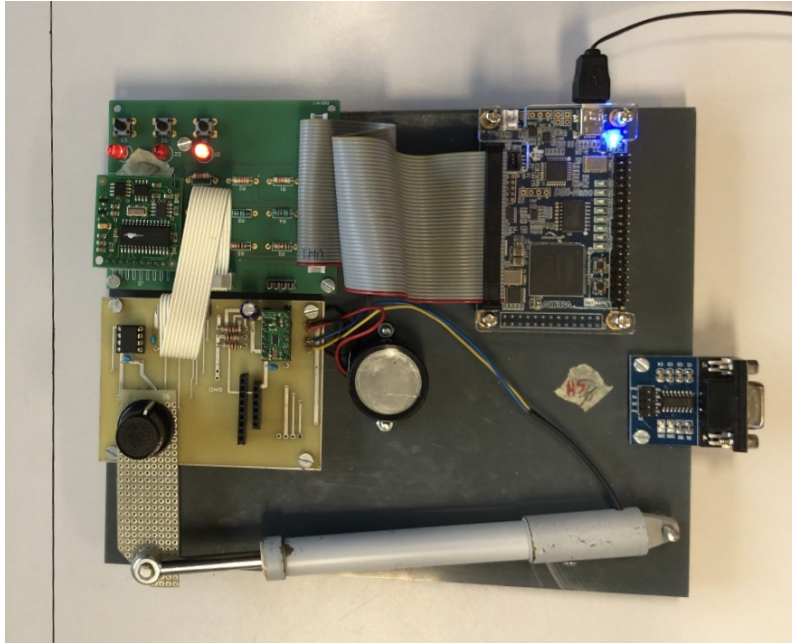


FIGURE 7 – Maquette validation projet

Nous avons réalisé trois tests pour tester le bon fonctionnement du projet. Pour le premier test nous avons utilisé un GBF qui nous a permis d'injecter directement sur le GPIO, du FPGA attribué, une fréquence qu'on peut modifier pour venir simuler la sortie de l'anémomètre. Après observation la fréquence est bien actualisée toutes les secondes comme demandé dans le cahier des charges. Pour le second test nous venons tester la fonction gestion vérin avec la fonction interface Homme système les boutons nous on permet de tester le bon fonctionnement qui contrôle le signal PWM du vérin. Pour la dernière fonction qui est l'asservissement du système complet nous observons un maintien du cap définit dans le logicielle téléversé sur la maquette de développement et que celle-ci respecte bien le cap fixé avant chaque passage en mode automatique.

## 6 Conclusion

Pour la réalisation du projet nous avons eu peu d'expérience dans le domaine de la programmation VHDL et avons dû nous former à l'aide d'exemples fournis durant la conception du projet. Grâce aux cartes de développement DE0 Nano nous avons pu nous entraîner et implémenter des fonctions en parallèle des cours et des séances de bureau d'étude.