

Mise en œuvre d'un pilote de barre franche pour voiliers

RAOUI Hafsa / ROBINSON Lucky
2022 / 2023

BE Synthèse et Mise en oeuvre des systèmes



Encadrant : M. Thierry PERISSE



Sommaire

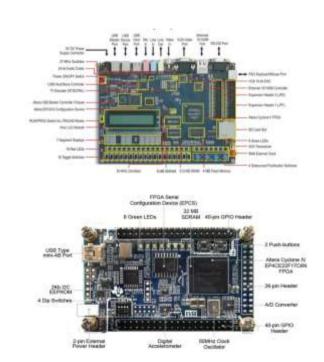
- 1. Introduction et présentation des outils
- 2. Analyse fonctionnelle et implémentation
- 3. Réalisation et test sur la carte
- 4. Conclusion

1. Introduction et présentation des outils

Pilote de barre franche pour voilier





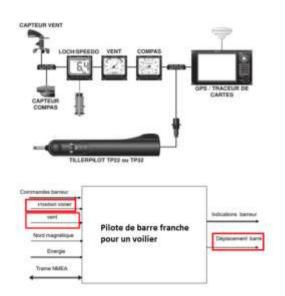


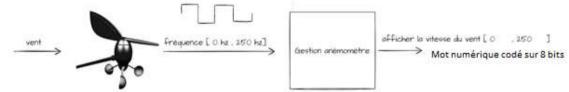


2. Analyse fonctionnelle et implémentation

- Présentation du système de pilote de barre franche pour voiliers
- Présentation de la fonction anémomètre



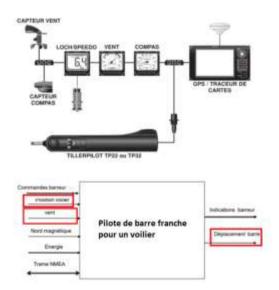


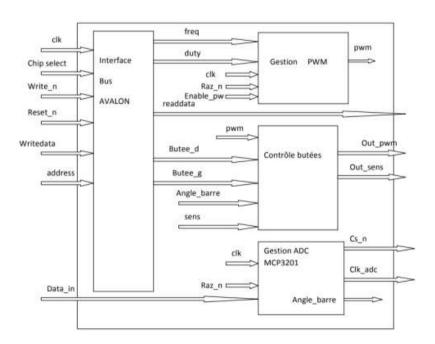


- Acquisition des informations relatives à la vitesse du vent par l'anémomètre
- L'anémomètre génère un signal carré de fréquence comprise entre 0 Hz à 250 Hz et le bloc fonctionnel affiche en mot numérique la vitesse allant de 0 à 250.

2. Analyse fonctionnelle et implémentation

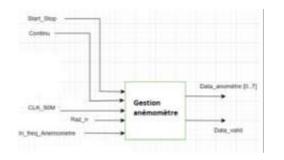
- Présentation du système de pilote de barre franche pour voiliers
- Présentation de la fonction vérin



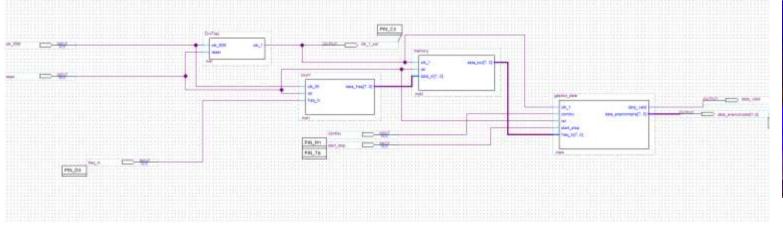




Bloc fonctionnel: Anémomètre



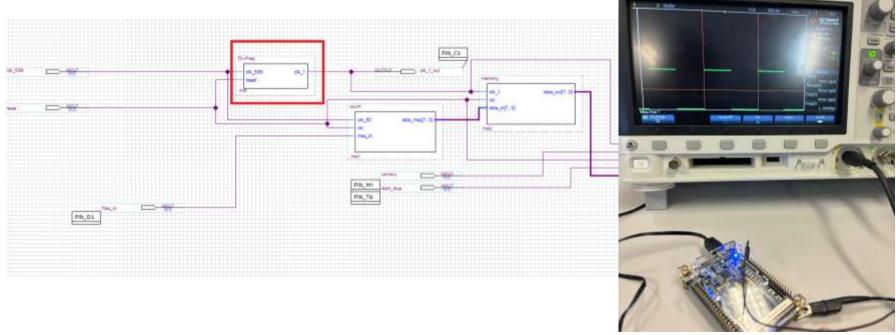
- Vue externe (entrées et sorties) et blocs fonctionnels de la gestion anémomètre
- Divisé en plusieurs blocs : la division de fréquence (50MHz à 1Hz), le comptage, la mémorisation et la gestion anémomètre





Frequence 8 Hz et 50 Hz

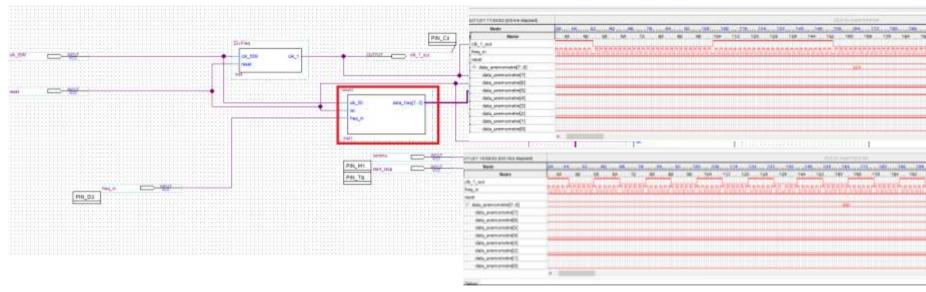
Sous bloc fonctionnel : la division de fréquence



- Entrées : signal horloge 50 MHz, reset et sortie : signal carré 1 Hz
- Implémentation d'un compteur qui s'incrémente à chaque front montant d'horloge
- Compteur = 25000000 : réinitialisation compteur et basculement de la sortie

X

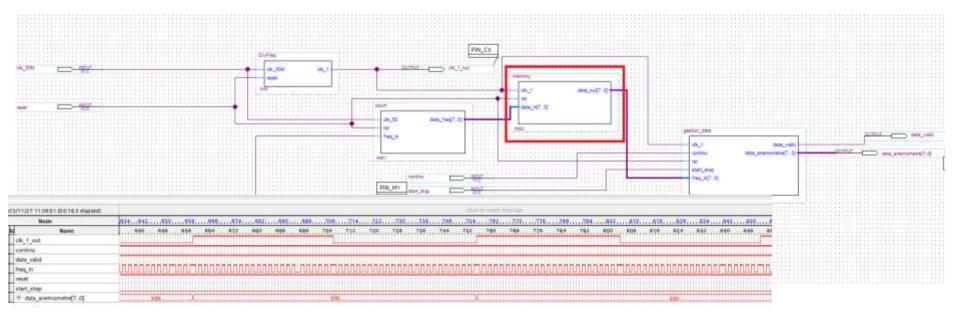
• Sous bloc fonctionnel : le comptage



- Entrées : signal 1Hz, signal reset, signal d'entrée de l'anémomètre et sortie : count (8 bits)
- Incrémentation d'un compteur : chaque détection de front montant de freq_in et le signal 1 Hz à l'état haut

X

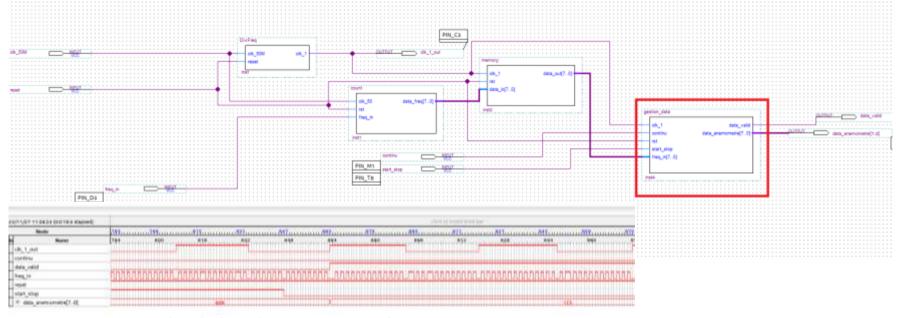
Sous bloc fonctionnel : mémoire



- Mémorisation de la donnée après la fin de l'état haut du signal 1Hz clk_1 Et actualisation de la donnée 8 bits en sortie chaque 1 seconde
- Le nombre de front montant pendant l'état haut ou bas du signal 1 Hz (demi-période) : correspond à la moitié de la valeur de data_out 8 bits.



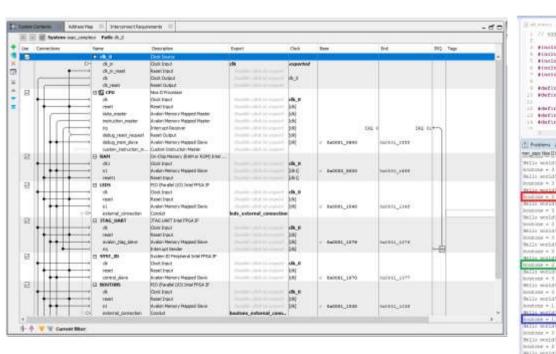
• Sous bloc fonctionnel : gestion donnée



- Deux modes de fonctionnement : continu ou mono coupe
- En continu : mise à jour de la donnée de vitesse toutes les secondes
- En mono-coupe : un signal start_stop qui commande l'affichage de donnée en sortie Si start_stop = 1 : acquisition de donnée, si start_stop = 0 : remis à 0 du signal Data_valid

Configuration SOPC : Affichage message et bouton poussoir





77 week writer last houses stadium "sysfelt sublo.n" PRINCIPLE OFFICERS Windleste Contratt, be #inclose "Alecca Arelen pro rege-b" // pro: fritty do contrigona les salement distinute des uniquitations // prior he function often edetion bootons (volatile oper "1 8007085 BASE | | #define leds (stationed int*) LEIS BASE if edefine freq cateligned but ") arminor ren i said II #define duty (whatgeed but *) (AVALOW DAME + 4) 11 #define control (unsigned int *) (AVALOR FRM) 1855 + 8) A femore of him D County # Non-A County II They was a female mor page Nos II Hardners configuration - cable: LSB States on localinat 1,68-4) de cos III: Lineaton III: Enumer ITAS JAST deg Metio world morning - 3 Mallo world Investment + 3 bootons - 1 Mallie world beatings - 3 Beday Midday booksta + 1 Settle world!

Salvery Married Williams

booksup + 1 Bellio weigh

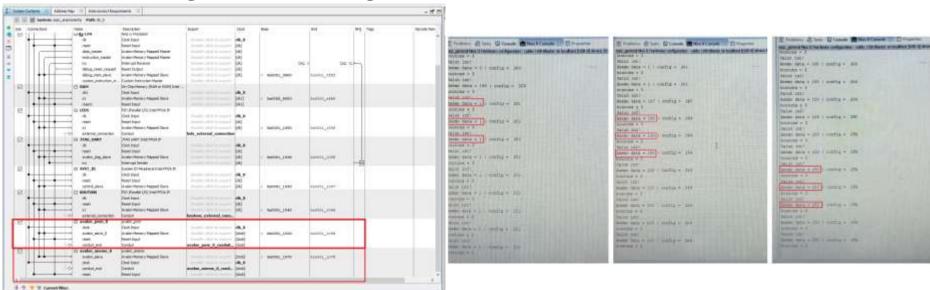
Soutple = 2

Beilly World! nestons < 1

- Travailler sur le plateform Designer pour ajouter les drivers utiles pour l'application
- NIOS II Eclipse IDE: code sur un fichier hello world.c et appel des adresses pour les boutons, leds en générant du BSP (Board Support Package)
- Affichage des commandes sur le terminal du NIOS II

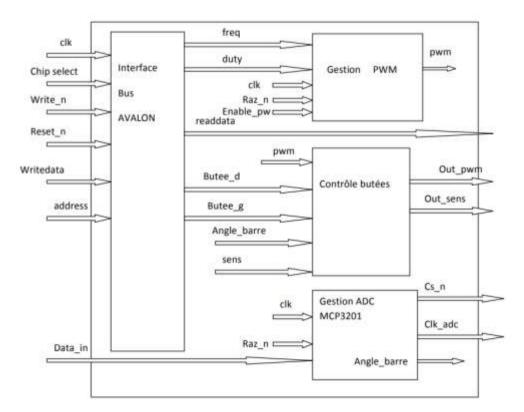


Configuration SOPC : intégration avalon PWM et avalon Anémomètre



Analyse fonctionnelle et implémentation

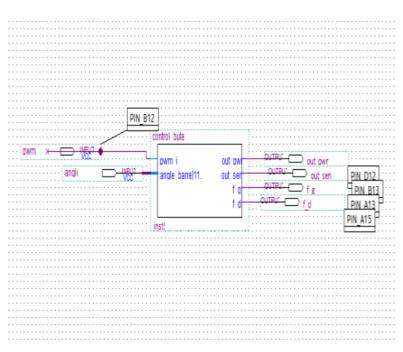
Présentation de la fonction vérin



La gestion de vérin constitué essentiellement de :

- La fonction Gestion PWM
- La fonction Gestion_butee
- La fonction Gestion_MCP3201
- La fonction d'interfaçage du bus Avalon

Sous bloc fonctionnel : Gestion de butés



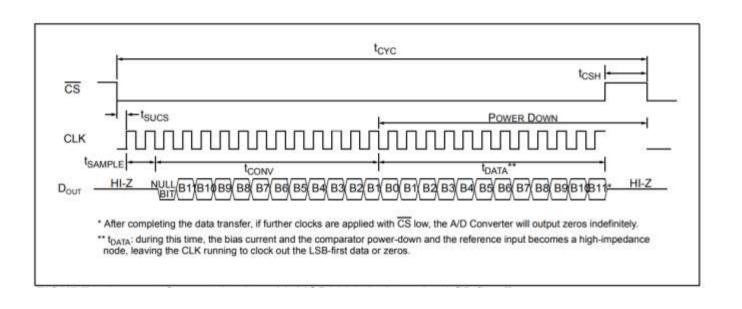
```
use IEEE, std logic 1164.all;
       use IEEE.std_logic_unsigned.all;
--use IEEE.std_logic_numeric.all;
     Bentity control_butee is
      Eport
            -- pwm_in, sens, enable: in std_logic;
            pwm in: in std logic:
            -butee_g, butee_d : in std_logic_vector (15 downto 0);
angle_barre : in std_logic_vector (11 downto 0);
out_pwm, out_sens, f_g, f_d : out_std_logic
12
13
14
       end entity:
     BARCHITECTURE arch of control_butee IS
signal sens : std_logic := '0'; -- Valeur fixee pour sens
signal enable : std_logic := '1'; -- Valeur fixee pour enable
             signal butee_g : std_logic_vector (15 downto 0) := "000010111011101000
signal butee_d : std_logic_vector (15 downto 0) := "00000000011001001
19
20
      FIBEGIN
     Eprocess (angle_barre)
23
24
25
       begin
26
                 f_g <= '1':
            elsif (angle_barre < butee_d and sens = '1' and enable ='1') then
29
30
                 out pam <= '0':
31
                 f_d <= '1';
32
33
                     out_pwm <= pwm_in;
34
                     f_g <= '0'
            end if:
       end process:
            out sens <= sens;
        END arch :
```

Sous bloc fonctionnel : Gestion de butés

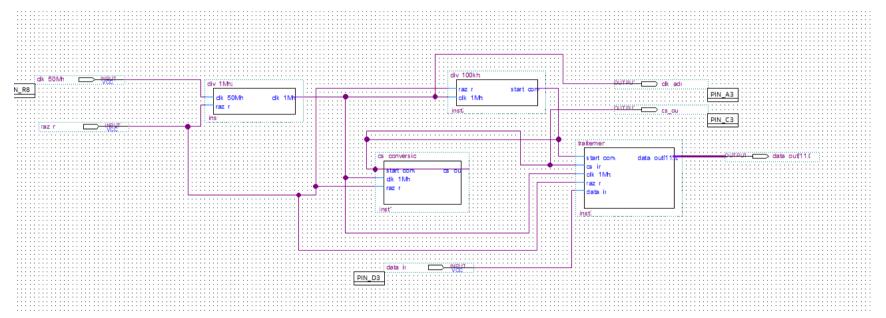




Sous bloc fonctionnel: Gestion du convertisseur AN MCP 3201



• Sous bloc fonctionnel: Gestion du convertisseur AN MCP 3201



- -Registre à décalage
- -Génération du 1MHz
- -Génération périodique

Sous bloc fonctionnel: Gestion du convertisseur AN MCP 3201



4. Conclusion



Merci pour votre attention!