

SOUTENANCE BE VHDL

PILOTE DE BARRE FRANCHE

MBOUNGOU & BA WAZIR

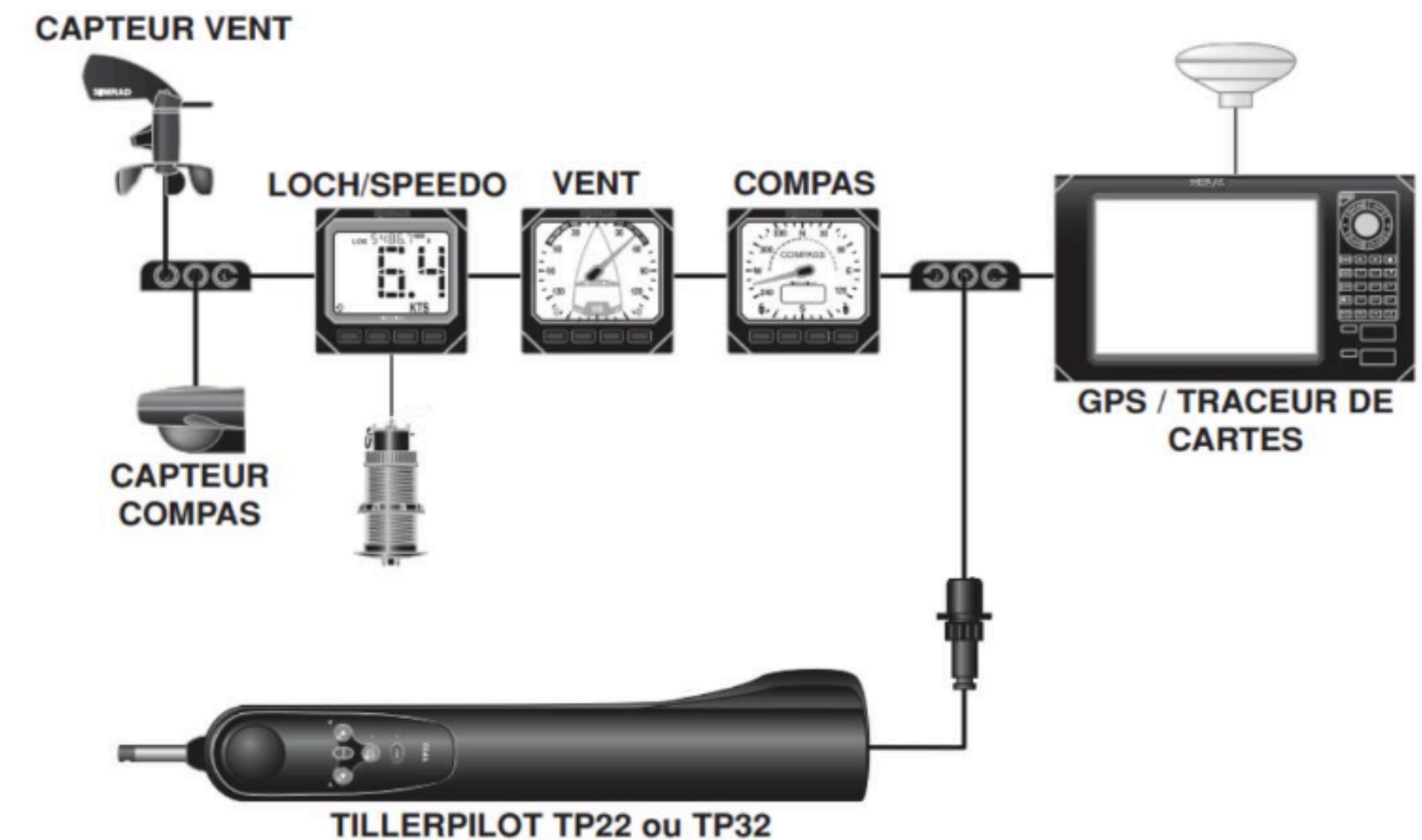
SOMMAIRE

- I - Conception du système
- II - Développements et simulations
- III - Intégrations des fonctions dans le SOPC

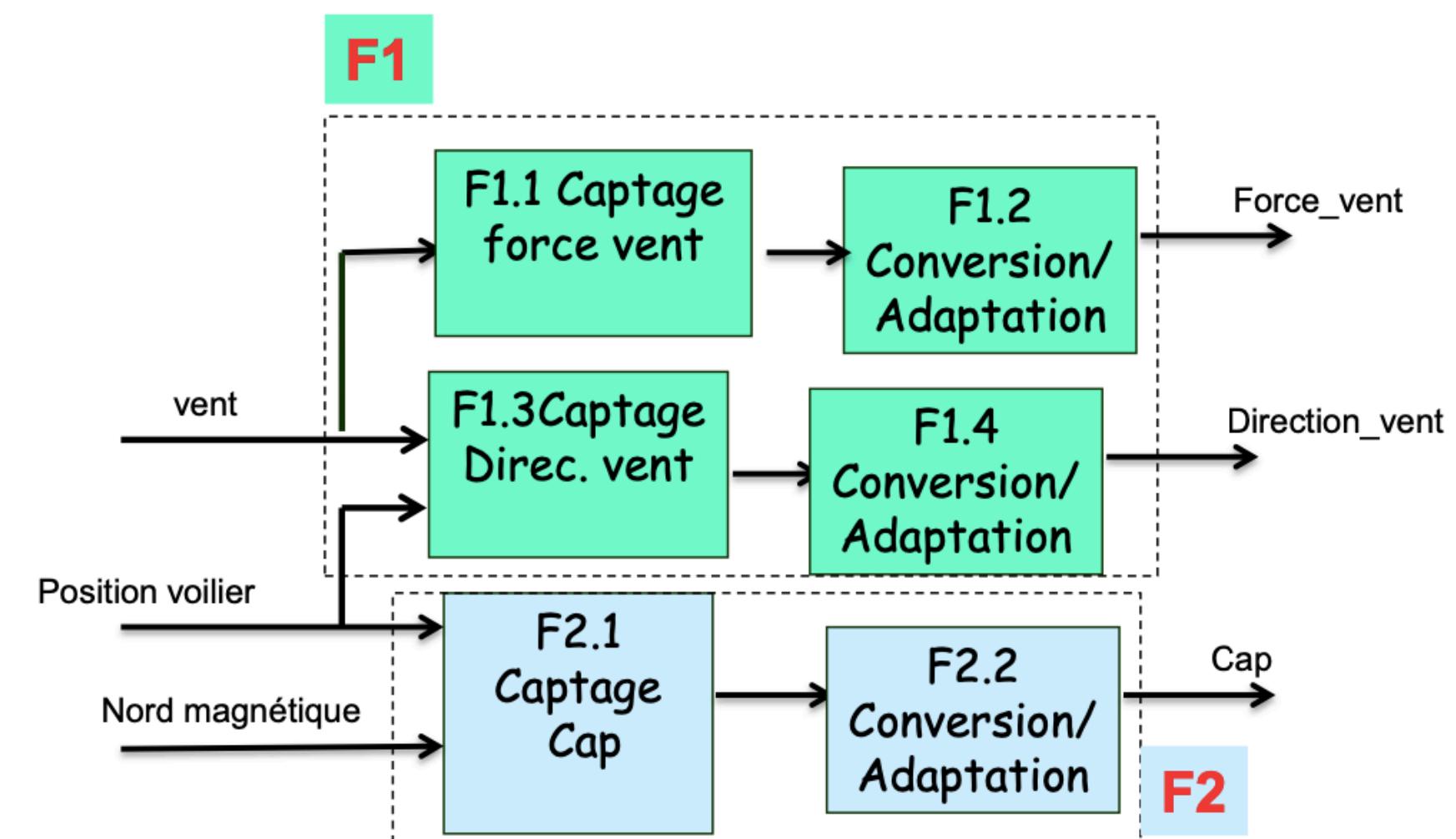
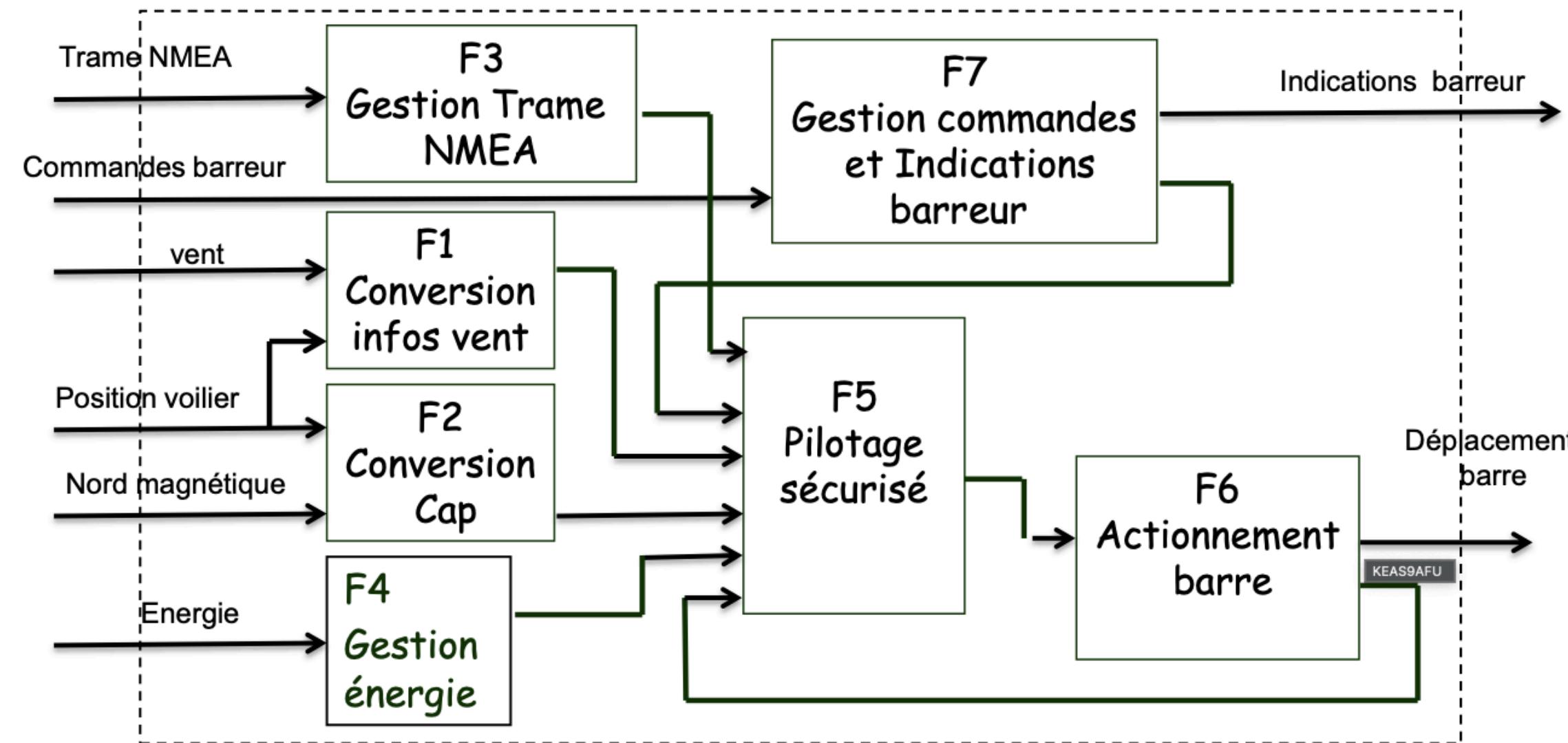
I - Conception

- Présentation du système

Un pilote de barre franche pour voiliers est un dispositif électronique conçu pour aider à maintenir le cap d'un voilier en utilisant une barre franche.



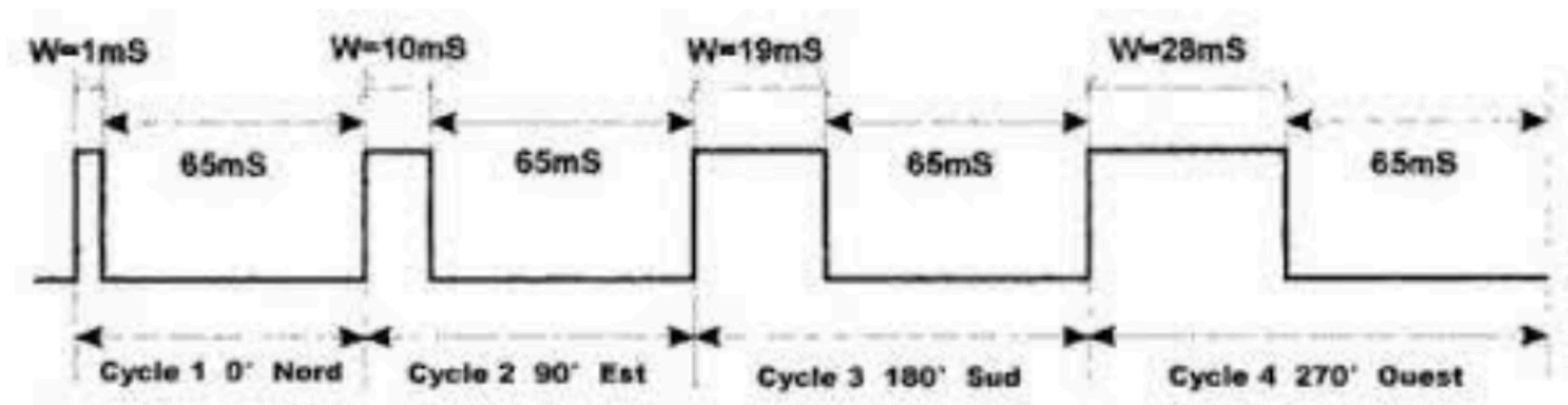
- Architecture
- Signal PWM
- Anémomètre
- Compas
- Gestion verin
- Interface Homme Machine



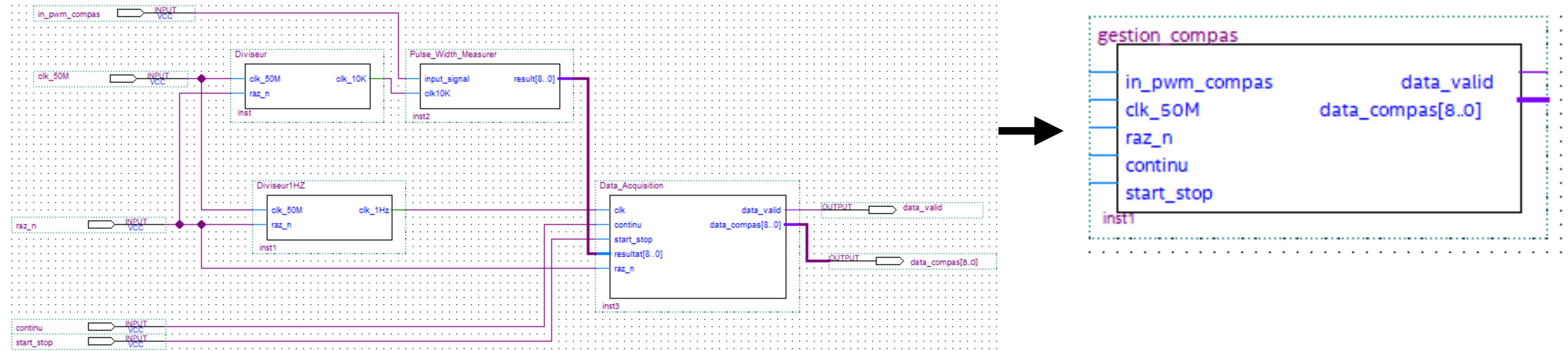
II - Développements et simulations

- Fonction simple : **COMPAS**

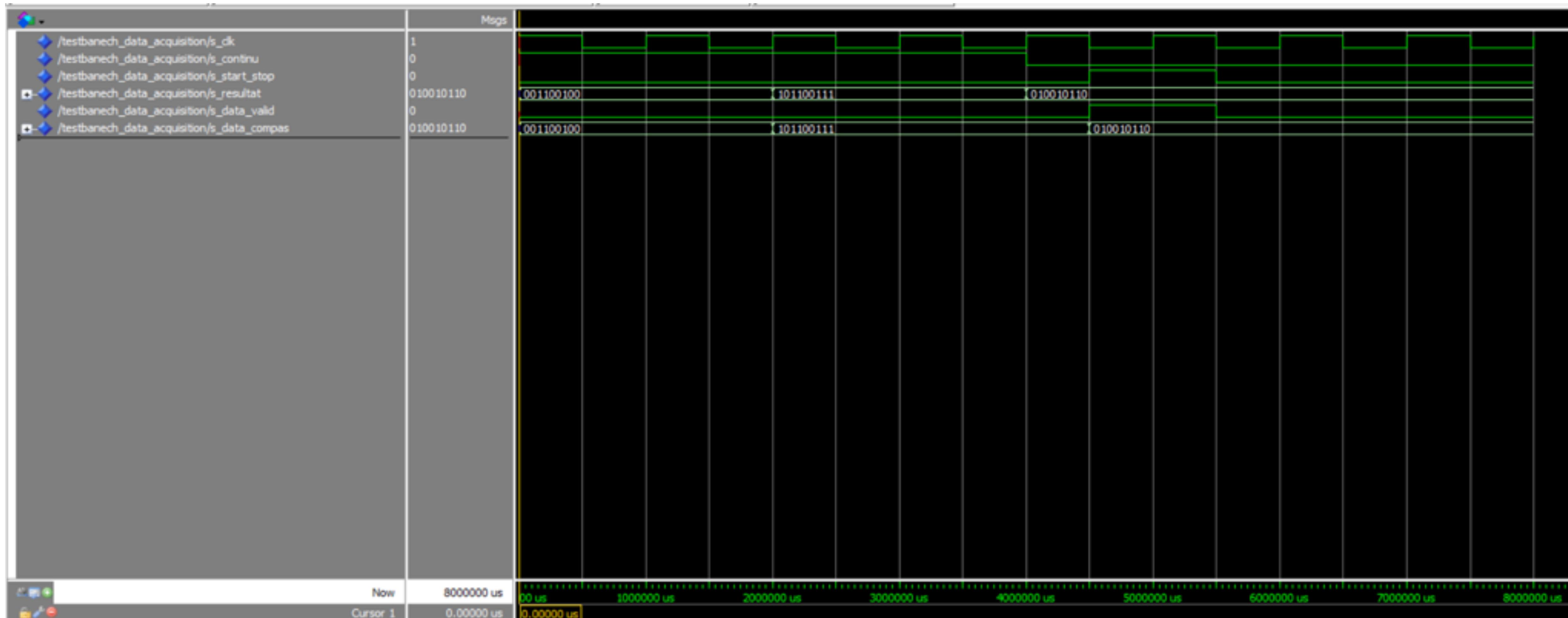
Le module compas a pour but de récupérer des mesures d'angles afin de fixer le cap.



- Analyse fonctionnelle

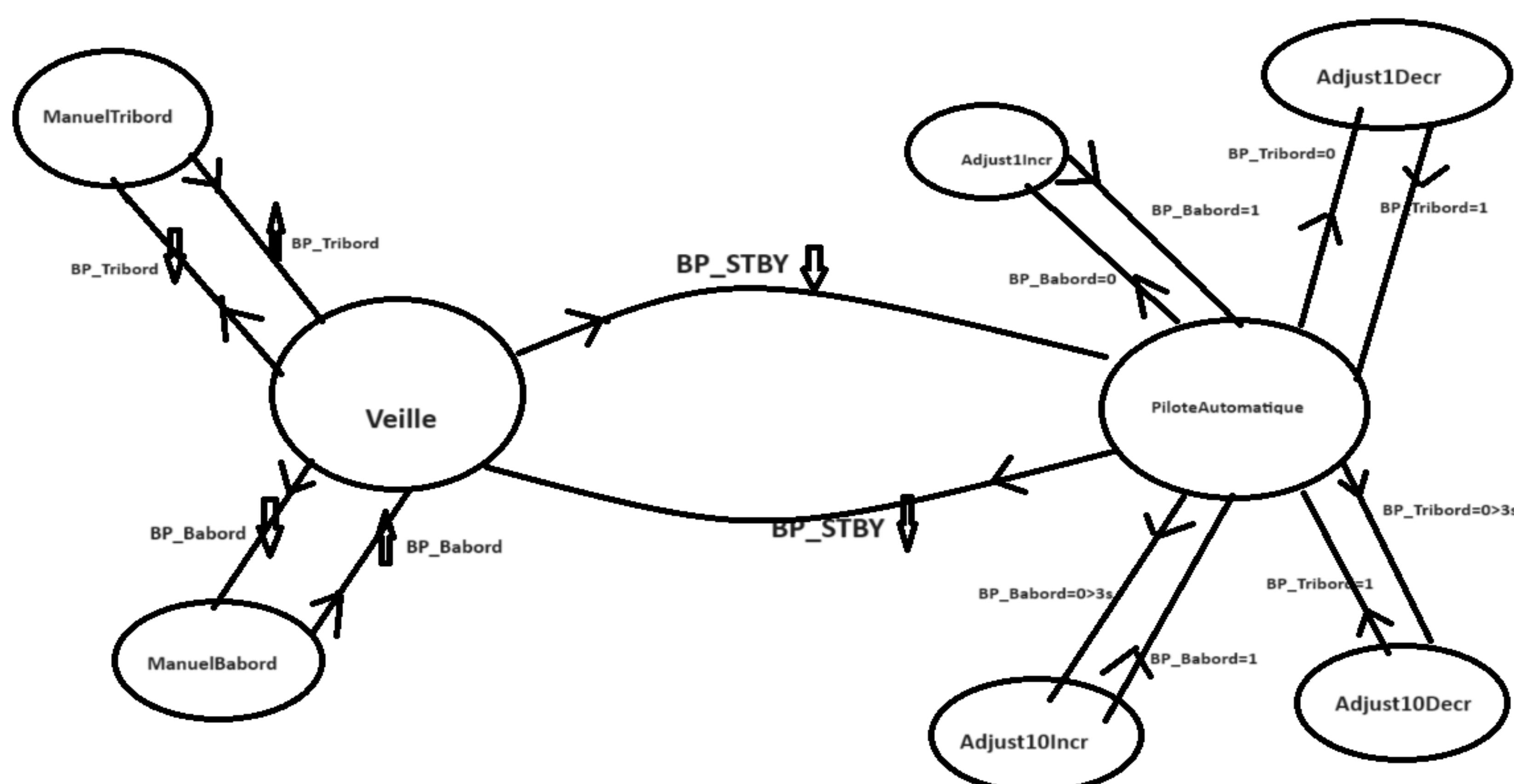


- Simulation du bloc D'acquisition

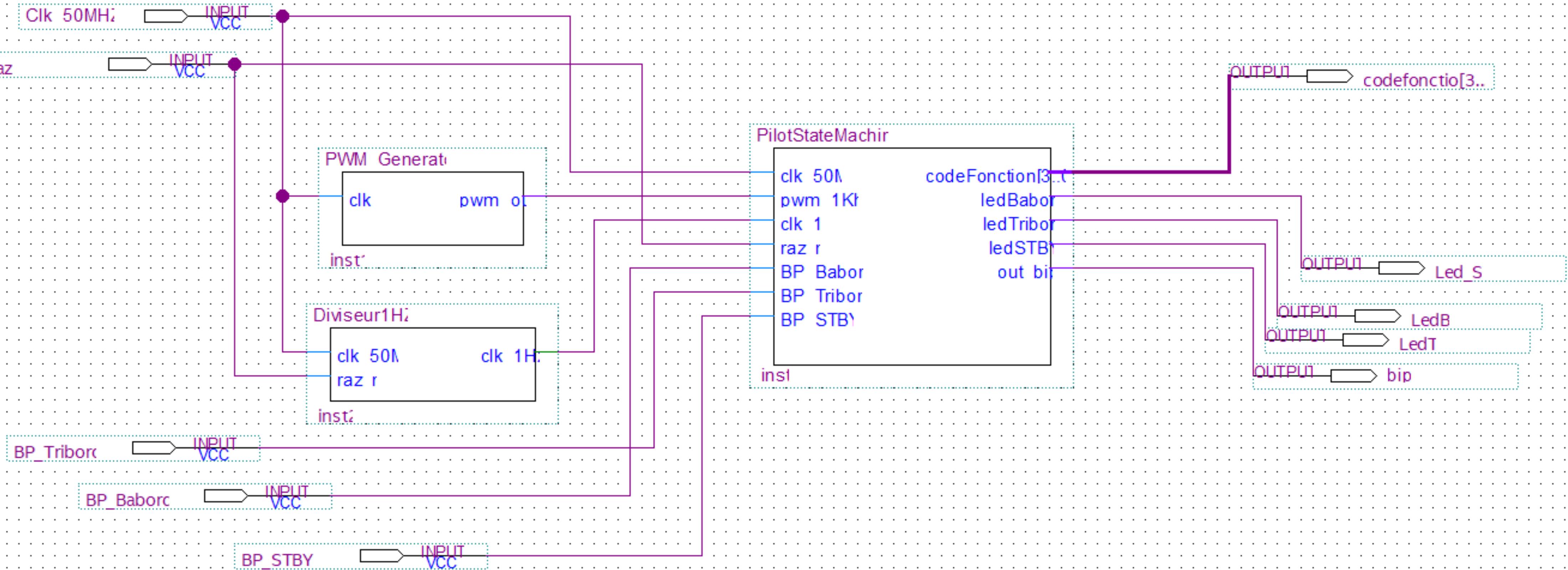


- Fonction complexe : **IHM**

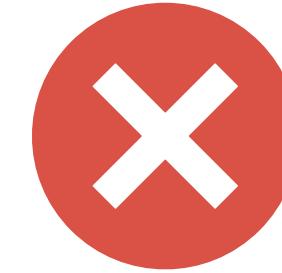
L'interface Homme Machine va permettre la mise en place du pilotage automatique de la barre franche pour les pilotes Simrad TP10, TP22 et TP32 sous forme d'une machine d'état.



- Analyse fonctionnelle



- Erreur sur la simulation



```
[1] 1072 VHDL Process Statement warning at PilotStateMachine.vhd(7), signal pmlm_time is read inside the process statement but not in the process
[2] 10536 VHDL Loop Statement error at PilotStateMachine.vhd(80): loop must terminate within 10,000 iterations
[3] 10441 VHDL Process Statement error at PilotStateMachine.vhd(40): Process Statement cannot contain both a sensitivity list and a Wait Statement
```

III - Intégration des fonctions sur le SOPC

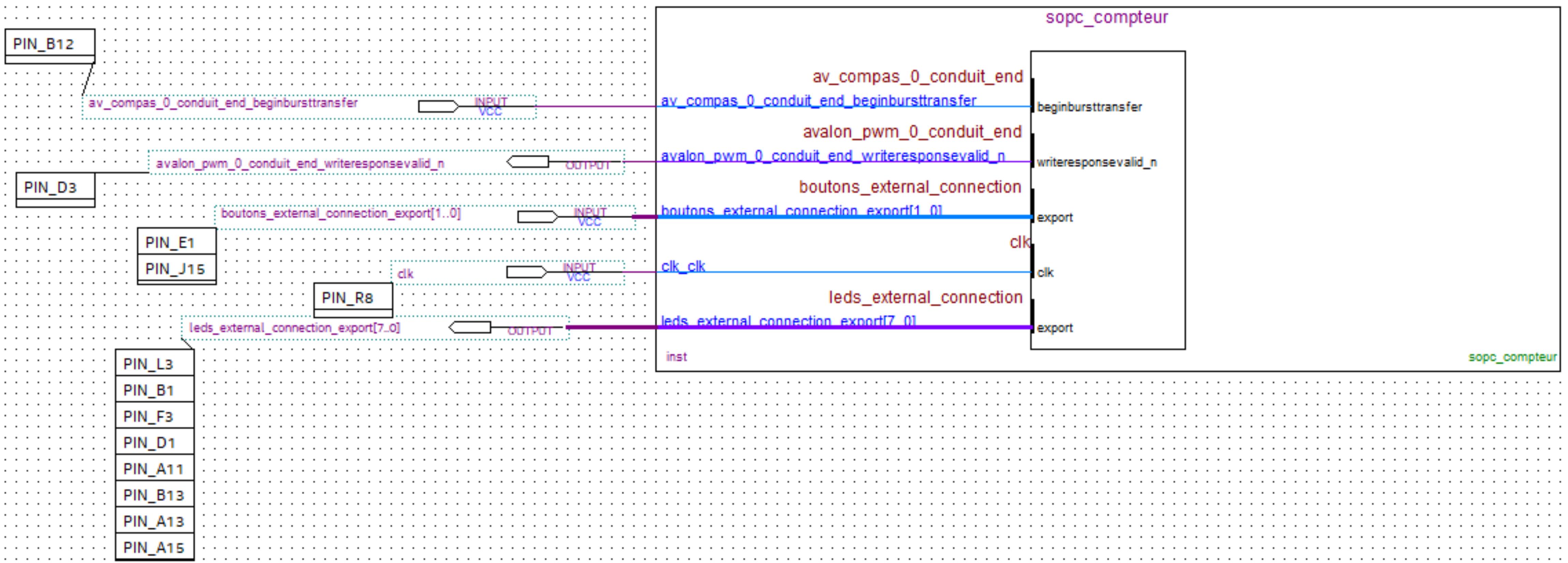
- Conception

AVALON_COMPAS.VHD



Use	C...	Name	Description	Export	Clock	Base
<input checked="" type="checkbox"/>		clk_0	Clock Source		exported	
<input checked="" type="checkbox"/>		CPU	Nios II Processor		clk_0	0x0001_0800
<input checked="" type="checkbox"/>		RAM	On-Chip Memory (RAM or ROM) Intel ...	clk_0	clk_0	0x0000_8000
<input checked="" type="checkbox"/>		LEDS	PIO (Parallel I/O) Intel FPGA IP	Double-click to export	clk_0	
		clk	Clock Input	Double-click to export	[clk]	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0001_1050
		external_connection	Conduit	leds_external_connection		
<input checked="" type="checkbox"/>		JTAG_UART	JTAG UART Intel FPGA IP	Double-click to export	clk_0	
		clk	Clock Input	Double-click to export	[clk]	
		reset	Reset Input	Double-click to export	[clk]	
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0001_1098
		irq	Interrupt Sender	Double-click to export	[clk]	
<input checked="" type="checkbox"/>		SYST_ID	System ID Peripheral Intel FPGA IP	Double-click to export	clk_0	
		clk	Clock Input	Double-click to export	[clk]	
		reset	Reset Input	Double-click to export	[clk]	
		control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0001_1090
<input checked="" type="checkbox"/>		BOUTONS	PIO (Parallel I/O) Intel FPGA IP	Double-click to export	clk_0	
		clk	Clock Input	Double-click to export	[clk]	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0001_1040
		external_connection	Conduit	boutons_external_connection		
<input checked="" type="checkbox"/>		avalon_pwm_0	avalon_pwm	Double-click to export	clk_0	
		clock	Clock Input	Double-click to export	[clock]	
		avalon_slave_0	Avalon Memory Mapped Slave	Double-click to export	[clock]	
		reset	Reset Input	Double-click to export	[clock]	
		conduit_end	Conduit	avalon_pwm_0_conduit_end	[clock]	
<input checked="" type="checkbox"/>		av_compas_0	av_compas	Double-click to export	clk_0	
		clock	Clock Input	Double-click to export	[clock]	
		avalon_slave_0	Avalon Memory Mapped Slave	Double-click to export	[clock]	
		reset	Reset Input	Double-click to export	[clock]	0x0001_1060

- Réalisation



```

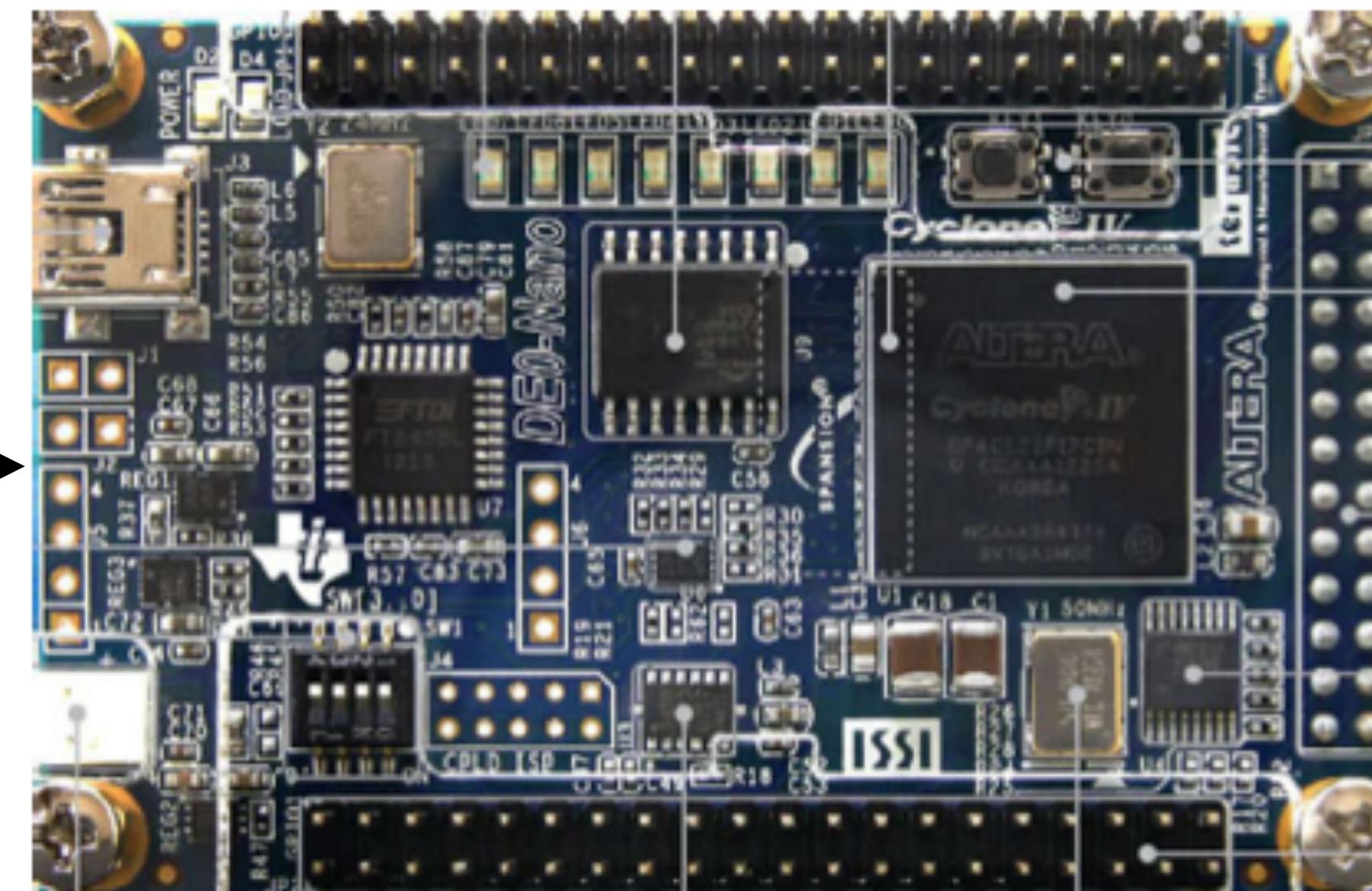
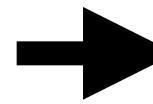
#define duty (unsigned int *) (AVALON_PWM_0_BASE + 4)
#define control (unsigned int *) (AVALON_PWM_0_BASE + 8)

unsigned int a;
int main()
{
    *freq = 0x0400; // divise clk par 1024
    *duty = 0x0200; // RC = 50
    *control = 0x0003;
    alt_putstr("Salut ext!\n"); // test si communication OK

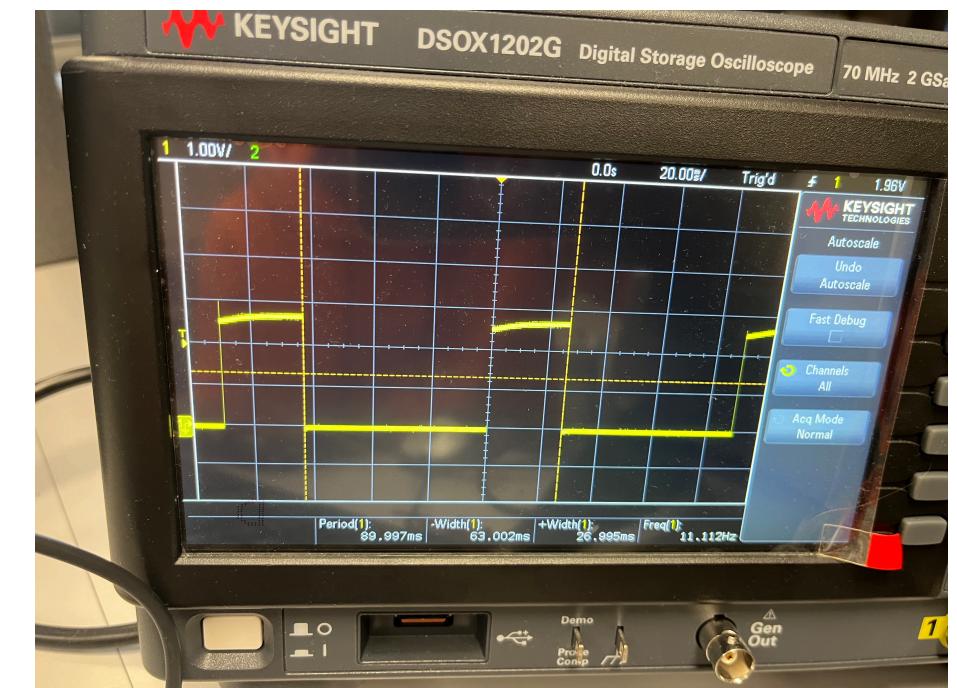
    while (1)
    {
        alt_putstr("Salut int!\n"); // test si communication OK
        /*leds = *boutons;
        a = *boutons & 3;
        printf("boutons = %d \n", a);
        usleep(1000000);
        switch(a)
        {
            {
                case 0 : *leds=0; break;
                case 1 : *leds=0; break;
                case 2 : break;
                case 3 : *leds=*leds + 1; break;
                default : *leds = 0; break;
            }
        }
        /*leds = 0xF;
        //usleep(1000000);
        /*leds = 0xF0;
        //usleep(1000000);
    }
    return 0;
}

```

Eclipse



Carte DEO nano



Conclusion

- Réalisation des étapes de conception d'un FPGA
- Prise en main de nouveaux outils numériques : VHDL, Quartus,...
- Compétences sur les projets futurs

Merci pour votre attention !