MINISTÈRE
DE L'ÉDUCATION
NATIONALE
ET DE LA JEUNESSE
Liberté
Égalité
Fraternité

EDE NUM 1

SESSION 2024

CAPET
CONCOURS EXTERNE ET CAFEP CORRESPONDANT ET TROISIEME CONCOURS

Section : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR

Option : INGÉNIERIE INFORMATIQUE

ÉPREUVE ÉCRITE DISCIPLINAIRE

Durée : 5 heures

Calculatrice autorisée selon les modalités de la circulaire du 17 juin 2021 publiée au BOEN du 29 juillet 2021.

L'usage de tout ouvrage de référence, de tout dictionnaire et de tout autre matériel électronique est rigoureusement interdit.

Il appartient au candidat de vérifier qu'il a reçu un sujet complet et correspondant à l'épreuve à laquelle il se présente.

Si vous repérez ce qui vous semble être une erreur d'énoncé, vous devez le signaler très lisiblement sur votre copie, en proposer la correction et poursuivre l'épreuve en conséquence. De même, si cela vous conduit à formuler une ou plusieurs hypothèses, vous devez la (ou les) mentionner explicitement.

NB : Conformément au principe d'anonymat, votre copie ne doit comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé consiste notamment en la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de la signer ou de l'identifier.
Le fait de rendre une copie blanche est éliminatoire.

Tournez la page S.V.P.

INFORMATION AUX CANDIDATS

Vous trouverez ci-après les codes nécessaires vous permettant de compléter les rubriques figurant en en-tête de votre copie

Ces codes doivent être reportés sur chacune des copies que vous remettrez.

► **Concours externe du CAPET de l'enseignement public** :

Concours	Section/option	Epreuve	Matière
EDE	14 13 E	101	9311

► **Concours externe du CAFEP/CAPET de l'enseignement privé** :

Concours	Section/option	Epreuve	Matière
EDF	14 13 E	101	9311

►**Troisième concours externe du CAPET de l'enseignement public :**

Concours	Section/option	Epreuve	Matière
EDV	14 13 E	101	9311

DOCUMENTS :

- DOCUMENTS TECHNIQUES :

  DT1 – Statistiques station météo

  DT2 – Documentation Vector3 Unity 3D

  DT3 – Modbus over TCP/IP

  DT4 – Protocoles usuels

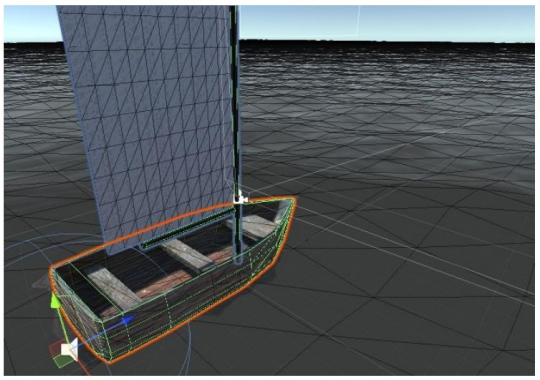
  Documents relatifs au support de l'étude

- DOCUMENTS RÉPONSES : DR 11, 37, 44, 45, 46

  Documents à compléter et à rendre par le candidat

IMPRIMERIE NATIONALE – 24 0233 – D'après documents fournis

SIMULATEUR DE VOILIER



À l'origine du projet se trouve la société AVICENNE et son dirigeant passionné de navigation à voile. Vivant et travaillant dans le sud de la Seine-et-Marne, il lui était difficile de rencontrer des conditions de navigation variées telles que celles qu'il rencontrait en mer. Ainsi décida-t-il de faire construire un simulateur de dériveur pour favoriser la formation des skippers et autres pilotes plaisanciers. L'idée commerciale fut de créer un prototype et, une fois ce prototype viable, d'industrialiser une solution dérivée de celui-ci.

Le sujet comporte une étude en cinq parties. Bien que les parties soient indépendantes, il est tout de même préférable de traiter la partie 1 en premier.

La **partie 1** introduit le contexte du système et étudie quelques liens entre la partie opérative et l'armoire de puissance. La **partie 2** modélise la poussée d'Archimède dans le moteur de jeu vidéo. La **partie 3** s'intéresse à la création de la consigne de vitesse en fonction des différents paramètres (vent, angle du vent, position bôme…), la **partie 4** étudie la communication réseau entre le jeu vidéo « Roll a Boat » et l'armoire de puissance. La **partie 5** conclut le sujet par une étude sur la mise en exploitation du simulateur.

Toutes les réponses devront être détaillées sur la copie et les résultats encadrés ou soulignés.

Préciser les unités des résultats.

Tous les documents réponses, complétés ou non, sont à rendre avec les copies.

Tournez la page S.V.P.

Partie 1 : Présentation du système et Impact socio-économique

Objectifs : *valider et Appréhender l'intérêt socio-économique d'un tel système et comprendre les mouvements d'un voilier en mer et sa simulation.*

1.1. Comparaison entre l'utilisation d'un simulateur et une école de voilier en bord de mer

Le simulateur a besoin d'un moniteur pour prodiguer des conseils aux stagiaires et assurer la sécurité de ceux-ci. Dans l'hypothèse où le moniteur est salarié d'une structure son contrat devrait être à plein temps de 35h par semaine avec 5 semaines de congés payés. Le simulateur est utilisable 35h par semaine et 47 semaines par an, il n'y a pas de contraintes de météo ou d'éclairement.

L'école de voile en bord de mer peut paralléliser jusqu'à 4 bateaux avec 1 moniteur, mais les bateaux ne sont utilisables que lorsque la météo est favorable et qu'il fait jour. Il est à noter également que le moniteur utilise un bateau à moteur pour assurer l'encadrement et la sécurité de ses cours.

Q 1. En considérant que les conditions idéales pour l'enseignement de la voile sont un vent inférieur à 56km.h^{-1} et une navigation de jour. Utiliser le document **DT1** pour **proposer** un comparatif entre les deux solutions. **Conclure** sur l'intérêt de ce simulateur.

1.2. Modélisation du système

Soit la modélisation suivante :

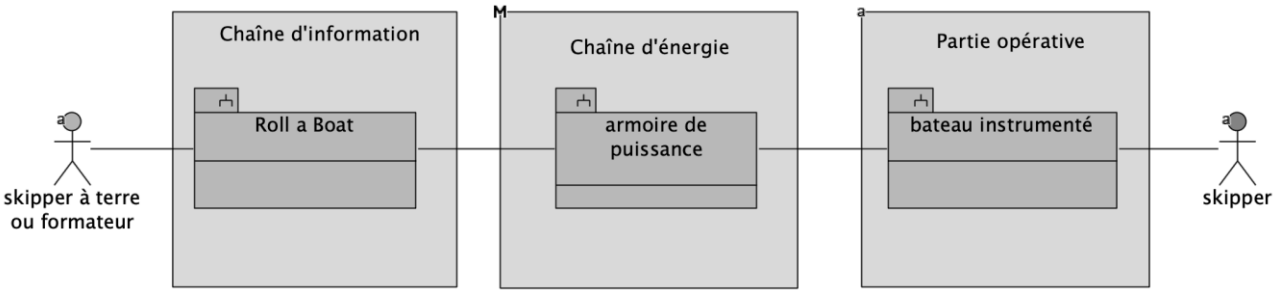


Figure 1 diagramme de contexte du système

Le skipper ici est celui qui apprend à naviguer, le skipper à terre apprend également à naviguer mais interagit avec le simulateur par un clavier. Le formateur a les mêmes outils que le skipper à terre. Le clavier peut être utilisé dans le bateau par le skipper. Le clavier permet de démarrer et d'arrêter une session de simulation.

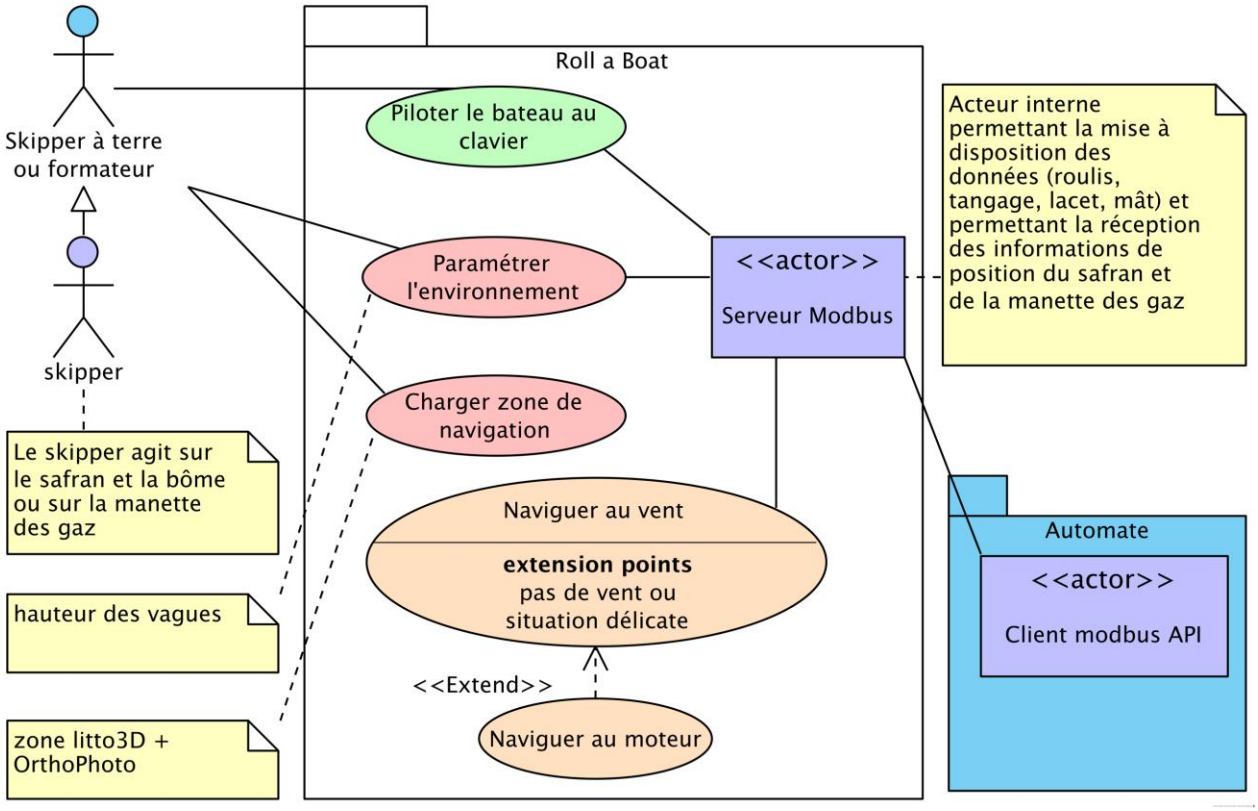


Figure 2 diagramme des cas d'utilisation partiel du logiciel « Roll a boat »

Q 2. **Expliciter** la relation entre les cas d'utilisation « Naviguer au vent » et « Naviguer au moteur ».

Q 3. **Expliciter** la relation entre les acteurs « Skipper à terre ou formateur » et « Skipper »

Le logiciel « Roll a Boat » est conçu afin de mettre à disposition les informations de simulation à l'automate. L'automate quant à lui, prend en charge le pilotage axe par axe (asservissements compris) de la partie opérative. La mise à disposition des données de simulation (roulis, tangage…) se fait par un serveur Modbus over TCP/IP.

La partie opérative agit sur la position des axes de roulis, tangage, lacet et bôme. La chaine de puissance n'est pas représentée ici. Les retours de position se font par des capteurs lus par l'automate.

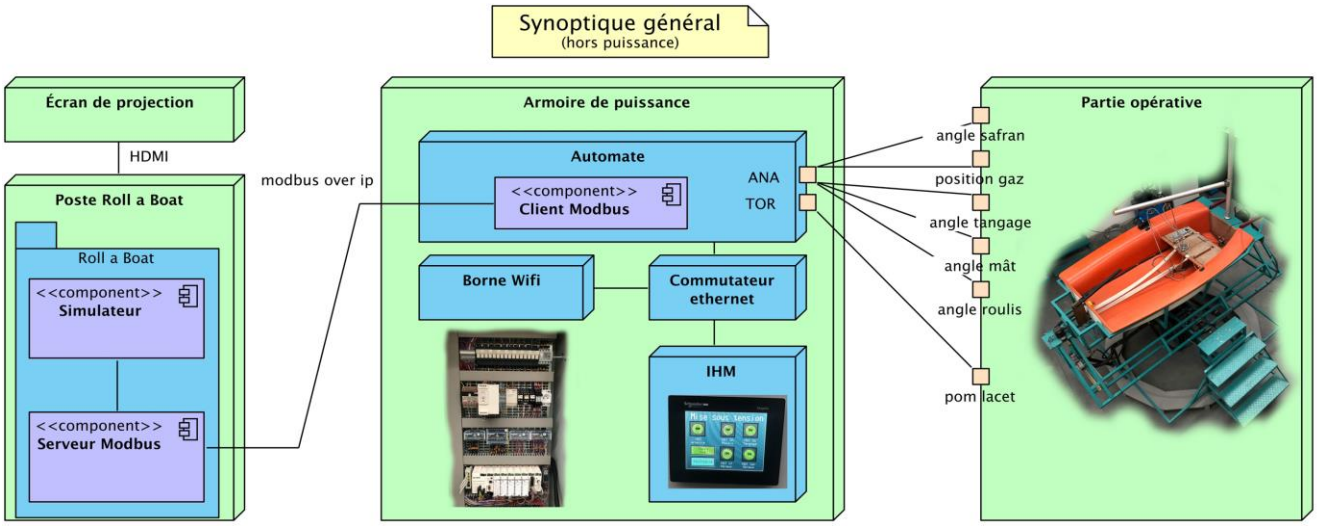


Figure 3 : diagramme de déploiement

Q 4. **Préciser** les supports physiques associés aux relations entre l'armoire de puissance et les autres composants du diagramme et **préciser** le nombre de ces composants.

1.3. Partie opérative

Le simulateur de bateau est constitué d'une coque de dériveur, équipée d'un mât et d'un safran. Cette coque est montée sur une structure métallique mobile qui permet de reproduire trois mouvements du bateau (roulis, tangage, rotation plane) ainsi que le retour d'effort sur le mât.

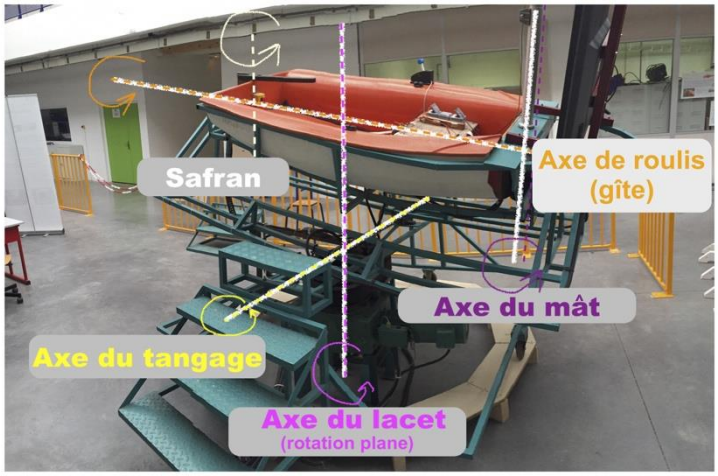


Figure 4 : Vue d'ensemble, identification des axes

Les mouvements de roulis, de tangage et le retour d'effort sur le mât sont assurés par des actionneurs hydrauliques commandés par des centrales hydrauliques munies d'électrovannes motorisées.

Le mouvement de rotation plane est assuré par l'intermédiaire d'une machine asynchrone commandée par un variateur de vitesse assurant la commande en position angulaire.

La partie opérative (Figure 4) comprend :

1 vérin hydraulique linéaire pour la commande du tangage	Sans référence
2 vérins rotatifs hydrauliques pour le roulis et le retour d'effort sur la bôme	Sans référence
1 machine asynchrone pour la commande de l'axe de lacet	P = 1,5 kW, cos φ = 0,84, I = 5,5 A
2 centrales hydrauliques équipées de 4 vannes proportionnelles	P = 1,5 kW chacune

L'armoire de commande comprend :

1 variateur de vitesse Leroy Somer	UMV 4301 2.5 T
1 automate programmable	M340
1 interface graphique	HMI STU 855

La partie opérative embarque également le poste informatique hébergeant l'application de simulation « Roll a Boat ».

Q 5. **Lister** les différences entre les mouvements possibles du simulateur et les mouvements réels d'un voilier ? Préciser les limites de ce simulateur.

Les axes du tangage et du roulis sont précisés dans le schéma cinématique ci-contre. L'axe du lacet n'y est pas présenté.
Le vérin permet de mouvoir le berceau autour du point A pour en assurer le tangage.
L3 correspond à la longueur du vérin et varie en fonction de la sortie de sa tige.
L_1 = 220cm ; L_2 = 170cm ; L_3 = 80cm mini à 140cm maxi

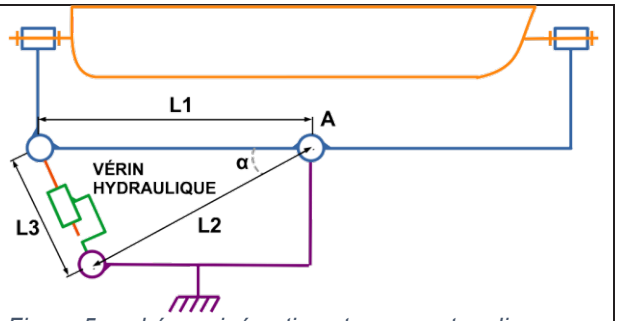


Figure 5 : schéma cinématique tangage et roulis

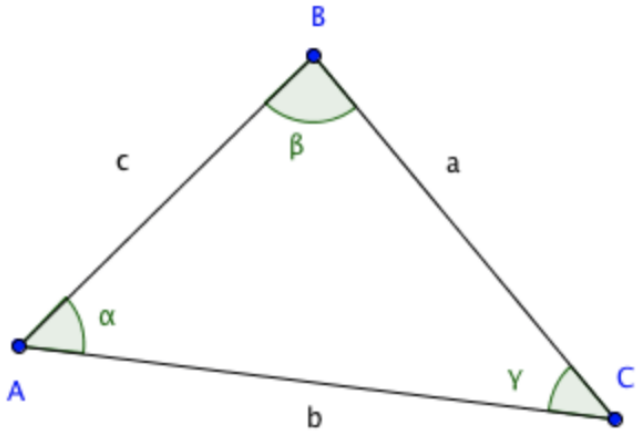


Figure 6 : Théorème d'Al-Kashi

$$a^2 = b^2 + c^2 - 2bc.\cos(\alpha)$$

$$b^2 = a^2 + c^2 - 2ac.\cos(\beta)$$

$$c^2 = a^2 + b^2 - 2ab.\cos(\gamma)$$

Q 6. **Exprimer** l'angle α de la Figure 5 en fonction de L1, L2 et L3 grâce au théorème d'Al-Kashi ;

Q 7. **Calculer** l'amplitude maximum de α entre les deux positions extrêmes du vérin.

Tournez la page S.V.P.

Le capteur de position utilisé au point A de la Figure 5 est un potentiomètre. Ce capteur est câblé comme ci-dessous.

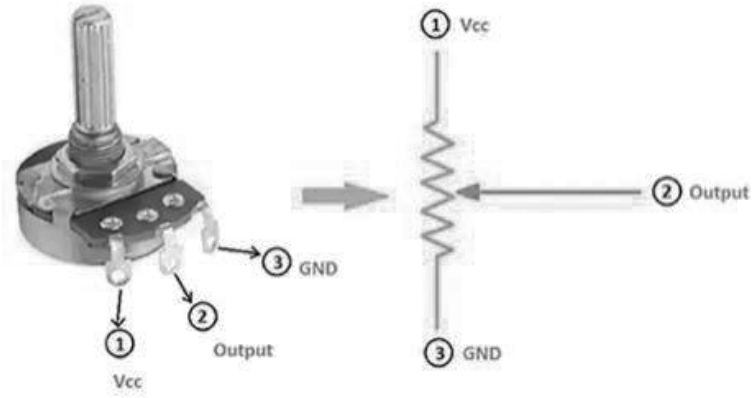


Figure 7 : câblage du potentiomètre de tangage

Q 8. **Donner** les tensions maximale et minimale entre les bornes 2 et 3 lorsque l'axe du potentiomètre est dans chaque position de butée.

Le tangage évolue de -10° à +10°. Le potentiomètre a une course maximum de 270°, il est installé en prise directe sur l'axe du tangage. Ce potentiomètre est alimenté entre 0 et 10v.

Q 9. **Calculer** l'amplitude de la variation de la tension sur la plage de déplacement du tangage.

L'entrée de l'automate possède des entrées analogiques, leur résolution est de 12 bits.

Q 10. **Calculer** le plus petit angle mesurable sur l'axe du roulis. **Préciser** si cet angle est suffisant pour cette application.

Le programme simulateur nommé *RollABoat* calcule les différentes commandes à appliquer à la partie opérative du simulateur à chaque image. L'évolution du simulateur est paramétrée par les actions du skipper sur le bateau tel que les actions sur l'écoute, le safran et éventuellement la commande du moteur.

Q 11. En utilisant la liste des actions ci-dessous, **compléter** le diagramme de séquence **DR11** qui indique les relations entre le skipper et le système dans le fonctionnement normal.

Liste des actions pour compléter :

- **Commander_le_bateau()** ;
- **Manœuvrer la manette des gaz** s'il n'y a pas de vent ou dans une situation délicate ;
- **Tirer sur l'écoute** pour manœuvrer la voile ;
- **Afficher l'IHM** (Interface Homme Machine) ;
- **Manœuvrer le Safran** pour donner un cap ;

Le vent souffle sur une embarcation de manière uniforme. La vitesse du vent se nomme *True Wind Speed* (TWS). Dans *RollABoat* sa valeur est stockée dans le registre **cons-tws**. L'angle que le vent forme avec le navire se nomme *True Wind Angle* (TWA) (cf. Figure 8). Sans action du skipper sur l'écoute, la bôme d'un navire se comporte comme une girouette

et tourne jusqu'à atteindre la position angulaire TWA-180. La différence entre TWA-180 et la position réelle de la bôme est stockée dans le registre **cons-difbotwa**.

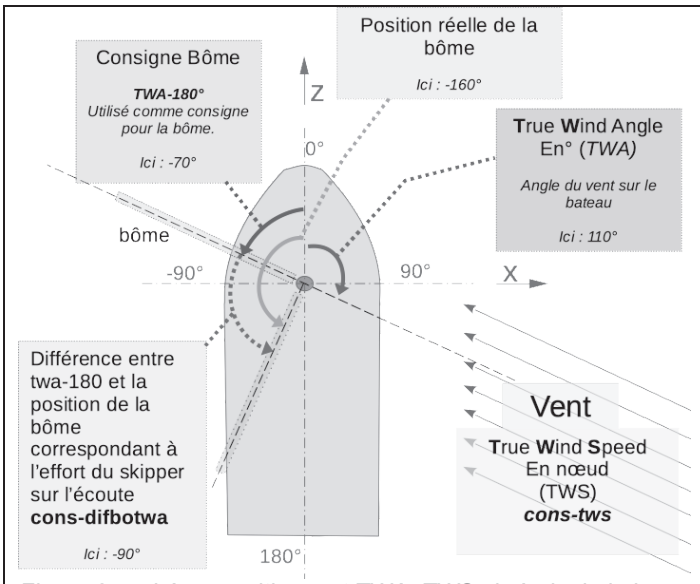
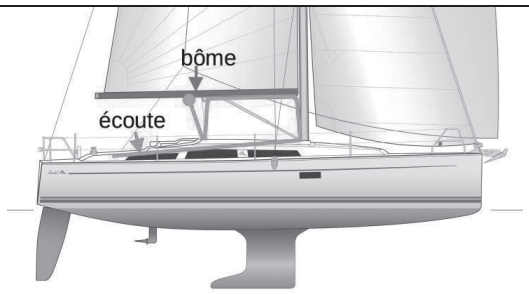


Figure 8 : schéma positionnant TWA, TWS vis à vis de la bôme



L'écoute est une corde (un bout dans le vocabulaire maritime). Elle permet d'orienter la bôme et donc la voile (à gauche ou à droite), plus on laisse de mou, plus la voile s'ouvre. On tire sur l'écoute pour lui donner un angle par rapport au vent.

L'automate utilise **twa-180°** comme consigne pour la position de la bôme sur la partie opérative (position d'équilibre).
L'asservissement du vérin rotatif du mât (déplaçant la bôme) est réglé « mou » de façon à permettre au skipper de tirer sur l'écoute pour écarter la bôme de sa position d'équilibre. Le skipper crée ainsi une erreur de position.

La puissance propulsive du vent utilisée par le voilier correspond au ratio (ou rapport) entre les positions de la bôme quand le skipper borde la voile (tire sur l'écoute) et choque la voile (relâche l'écoute) multiplié par la puissance maximum disponible. Cette puissance maximum est fonction de la vitesse du vent $\boldsymbol{P_{propulsive} = P_{Maximum}.ratio}$.

Selon la direction du vent on doit considérer deux cas :

- Vent venant de l'arrière du bateau, navigation « au large » : le maximum de puissance est obtenu quand |**cons-difbotwa|** est de 90° (cf. Figure 8).
- Vent venant de l'avant du bateau, navigation « au près » : le maximum de puissance est obtenu si |**cons-difbotwa|** est de 180° (la bôme est alignée sur 180°) ;

L'entreprise souhaite utiliser les profils de ratio suivant pour paramétrer la puissance propulsive à appliquer sur le bateau :

Tournez la page S.V.P.

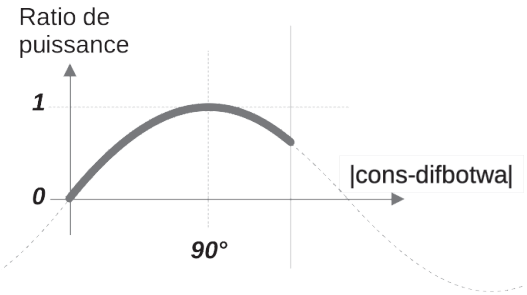


Figure 9 Navigation au large

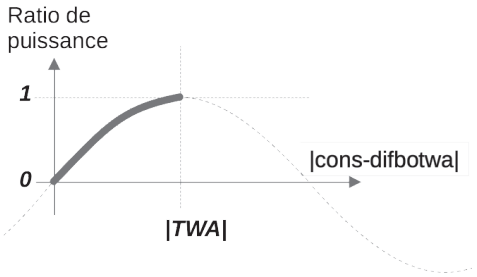


Figure 10 Navigation au près

Le point d'ancrage de l'écoute, à l'arrière du bateau, permet en tirant sur celle-ci d'aligner la bôme à 180° (Figure 8).

Q 12. Navigation « au large » : **Proposer** une expression du ratio décrit Figure 9 en fonction de cons-difbotwa. Cette expression fera intervenir une fonction sinusoïdale.

Q 13. Navigation « au près » : **Proposer** une expression du ratio décrit Figure 10 en fonction de cons-difbotwa. Cette expression fera intervenir une fonction sinusoïdale.

Q 14. **Proposer** une explication sur la signification des traits pointillés des courbes Figure 9 et Figure 10.

La rotation autour de l'axe vertical (lacet) est assurée par une machine asynchrone et son motoréducteur. Le retour de position se fait par un codeur incrémental le signal « pom lacet » est lié à ce codeur.

Q 15. **Que signifie** « pom » dans « pom lacet ». **Décrire** l'impact de ce type de capteur sur la mise en marche du système vis-à-vis d'un codeur absolu ?

Partie 2 : Poussée d'Archimède

Objectifs : *déterminer quelle poussée appliquer à un objet 3D pour simuler la poussée d'Archimède.*

L'application « Roll a Boat » est développée en C# sur Unity. Unity est un moteur de jeu développé depuis 2005 par la société Unity Technologies. La physique de la flottabilité des objets n'est pas incluse à Unity. Néanmoins, « Roll a Boat » intègre plusieurs notions pour l'aborder. Cette partie du sujet se limitera à l'application du théorème de la poussée d'Archimède.

Q 16. En supposant un cube de 50cm de côté. **Calculer** son volume en m^3.

Q 17. **Exprimer** quelle est sa masse maximum pour qu'il flotte.

Q 18. **Quelle est** l'intensité de la force de la poussée d'Archimède **et** la direction de celle-ci lorsque le cube est entièrement immergé ?

2.1. Objets maillés

La mer est simulée par un objet maillé plan sur lequel un ensemble de sinusoïdes est appliqué pour les vagues. La scène suivante, Figure 11, montre un prototype mettant en œuvre la poussée d'Archimède, elle est composée de :

- La mer, un objet maillé plat (sans vague ici) ;
- Le fond marin, un objet maillé plat également ;
- Le bateau, qui en première approche, est un cube.

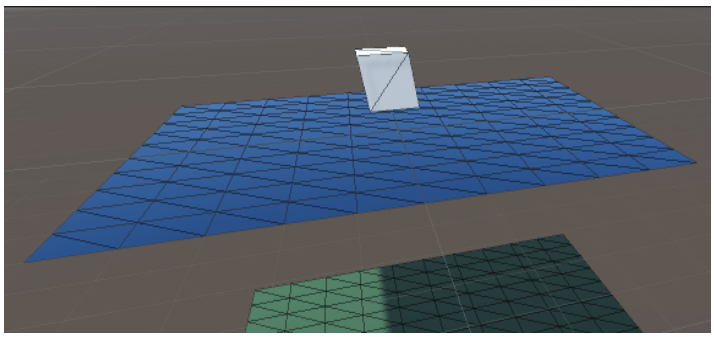


Figure 11 : Scène prototype pour la validation de la poussée d'Archimède

Chaque face du cube, Figure 11, est composée de 2 triangles et un cube est composé de 6 faces et 8 sommets.

Dans la suite on utilisera le logiciel Unity 3D, une documentation **DT2** est disponible.

```
Vector3[] vertices = new Vector3[4]
{
     new Vector3(0, 0, 0),
     new Vector3(width, 0, 0),
     new Vector3(0, height, 0),
     new Vector3(width, height, 0)
};
```

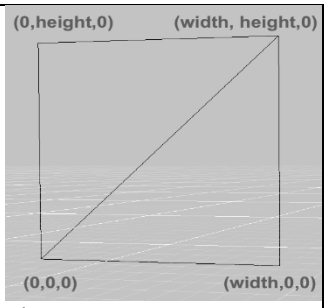


Figure 12 : Sommets

Ce premier extrait de code permet de créer une liste de 4 *vertices* (sommets) Figure 12.

```
int[] tris = new int[6]
{
     0, 3, 1, // lower right triangle
     0, 2, 3 // upper left triangle
};
```

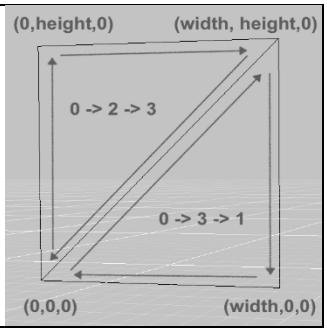


Figure 13 : Triangles

L'extrait ci-dessus permet de créer 2 triangles orientés avec le parcours des sommets dans le sens horaire, Figure 13. **0->3->1** devient **0,3,1** et **0->2->3** devient **0,2,3** dans le code de création des triangles. Dans la suite du sujet, l'ordre de parcours des sommets pour la création des triangles n'a pas d'importance.

Dans *Figure 12* et *Figure 13*, il y a donc 2 triangles, 4 sommets et 6 « chemins » formant un carré.

Q 19. Sur la base des exemples présentés ci-dessus **proposer** la création des sommets d'un cube utilisant les variables : hauteur ***height,*** largeur ***width*** et profondeur ***depth***.

Q 20. Sur la base des exemples présentés ci-dessus **proposer** la création des faces du cube utilisant les même variables (*height, width* et *depth*).

Deux opérations sont encore nécessaires pour créer un objet maillé complet sous Unity : la définition des normales pour permettre la gestion des ombrages et les coordonnées de textures pour permettre une bonne application de celles-ci (non présenté ici).

2.2. Calcul du volume sous l'eau

Pour calculer les forces à appliquer au bateau virtuel, il convient de déterminer quel est le volume situé sous l'eau. Le principe retenu est celui de mesurer le volume situé entre le triangle immergé et l'eau.

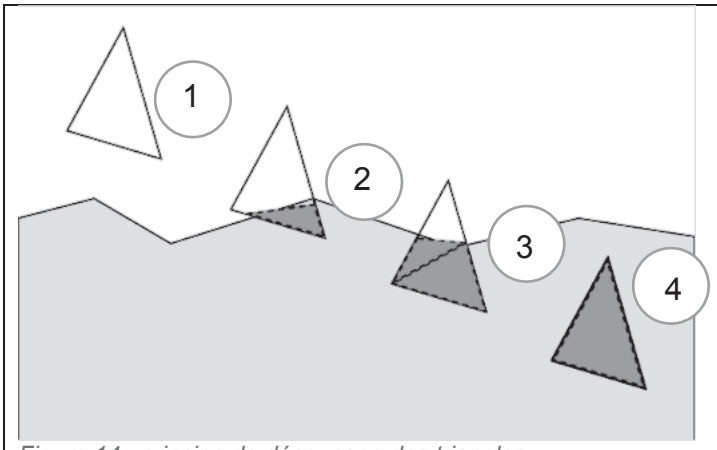
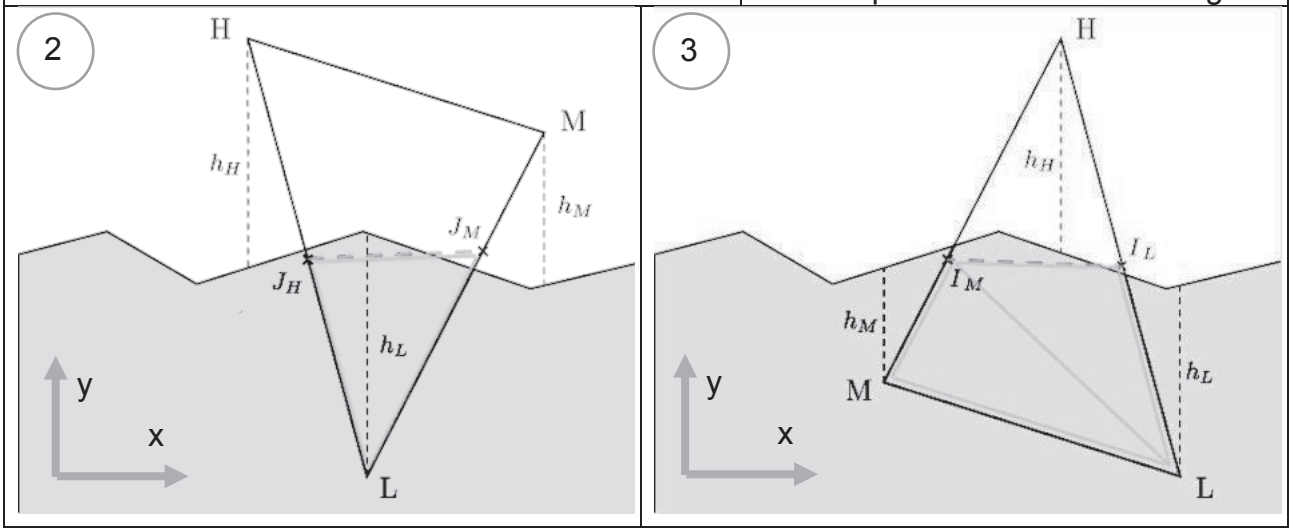


Figure 14 : principe de découpage des triangles

Chaque triangle est analysé sommet par sommet avec 4 cas possibles :

- **cas 1 :** Triangle au-dessus de l'eau,
- **cas 2 :** 1 sommet du triangle sous l'eau,
- **cas 3 :** 2 sommets du triangle sous l'eau,
- **cas 4 :** Triangle au-dessous de l'eau.

Dans le cas où le triangle est entre les deux (cas 2 et 3), il doit être redécoupé en deux ou trois triangles.



L'objectif est de redécouper les triangles qui sont en partie sous l'eau en un ou deux triangles qui eux, seront sous l'eau entièrement. Une méthode existe pour calculer la distance à l'eau, elle sera abordée dans la suite du document. Pour effectuer le découpage on va ajouter des points placés sur les segments existants. Le positionnement de ces points sur les segments se fait en fonction de la distance à l'eau h_{point}.

Étude de cas, le point J_H du **cas 2** ci-dessus :

> On ajoute un point J_H sur le segment HL en suivant la règle de positionnement suivante, le ratio $\frac{LJ_H}{\text{LH}}$ est égale à celui de $\frac{|h_L|}{|h_L|+|h_H|}$. Par exemple, si $h_H = 2$ et $h_L = -2$, J_H sera positionné au milieu de HL, soit un ratio de 50%. Si $h_H = 2$ et $h_L = -3$ alors J_H sera positionné à 3/5 de L, soit un ratio de 60%.

La documentation de la classe Vector3 est dans DT2, le calcul permettant de placer le point J_H sur le segment [HL] est traité dans les questions suivantes.

Hypothèse : H, L sont des points 3D stockés dans des Vector3 et h_H et h_L sont des variables de type float (h_H est positif et h_L négatif).

Q 21. **Proposer** la ligne de code permettant de créer le *Vector3 LH* représentant le vecteur $\overrightarrow{\text{LH}}$.

Q 22. **Proposer** la ligne de code permettant de calculer le ratio de la distance entre L et J vis-à-vis de LH.

Q 23. **Proposer** la ligne de code permettant de multiplier le *Vector3 LH* par le ratio et **créer** le *Vector3 LJ_H* représentant le vecteur $\overrightarrow{\mathrm{L}J_H}$

Q 24. **Proposer** la ligne de code permettant de créer le Vector3 J_H contenant les coordonnées du point J_H.

Q 25. **Proposer** un algorithme permettant de générer une liste de triangles sous l'eau conformément à la Figure 14 (on pourra proposer une solution en pseudo-code, python, JAVA, C/C++ ou C#).

Cet algorithme doit prendre en entrée les tableaux de triangles et sommets comme définis dans Figure 12 et Figure 13. En sortie l'algorithme doit créer deux nouveaux tableaux, sommets et triangles, issus des découpages tel que décrit Figure 14.

Une méthode « `distanceToWater(Vector3 vertice) : float` » existe pour obtenir la hauteur algébrique d'un sommet par rapport à l'eau.

Q 26. **Lister** les différents éléments restant à simuler et émettre des pistes pour le faire. **Conclure** sur la validité de l'approche.

Partie 3 : Polaire de vitesse

Objectifs : *déterminer à quelle vitesse le voilier doit se déplacer et quelle force de propulsion doit lui être appliquée.*

3.1. Polaire de vitesse

Les profils des voiles (aérodynamiques), la coque d'un voilier (hydrodynamique) et, dans une moindre mesure, son chargement déterminent le comportement d'un voilier. La polaire de vitesse ci-contre est un tracé réel de la vitesse du bateau pour plusieurs vitesses du vent, *TWS*. La vitesse du navire est représentée suivant l'angle du vent réel, *TWA* par rapport à la direction de déplacement du voilier de 0 degré (le vent arrive face au bateau) à 180 degrés (le vent arrive par l'arrière).

TWS (nœuds) / TWA (°)

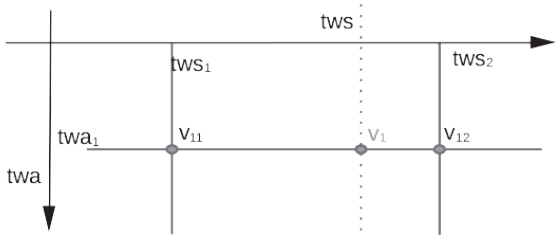
TWA (°) \ TWS (nœuds)	0	4	10	16	20	24	30
30		2.50	5.70	6.20	6.40	6.49	6.39
45		3.70	6.80	7.80	8.20	8.76	8.80
65		4.80	7.90	9.70	10.80	11.60	12.10
85		5.00	8.68	11.70	12.70	13.43	14.30
105		4.89	9.00	12.40	13.80	14.90	16.30
125		4.30	8.50	12.40	14.20	15.60	17.40
145		3.00	6.80	10.10	12.20	14.60	17.00
180		1.30	3.30	5.00	6.10	7.10	9.30

Pour un vent de 16 nœuds, un angle de vent de 125° le bateau (class40) se déplace au maximum à 12,4 nœuds. Le maximum est obtenu avec une voile parfaitement réglée.

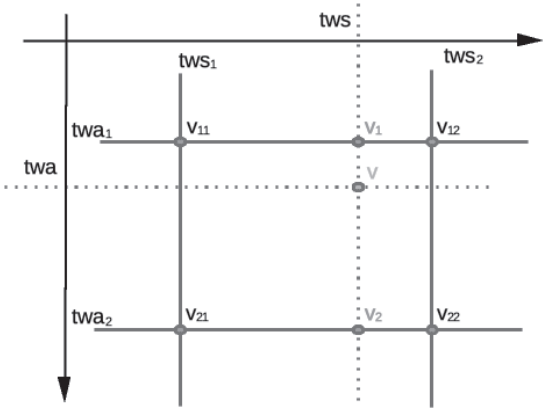
Q 27. **Retrouver** la vitesse du bateau pour un vent de 10 nœuds et un angle de 65°.

Le logiciel RollABoat utilise une interpolation bilinéaire pour approximer les valeurs manquantes.

Q 28. **Proposer** une équation exprimant V_1 en fonction de V_{11},V_{12}, *tws*, tws_1 et tws_2. **Appliquer** cette formule pour un vent de 12 nœuds et un angle à 85°.



Q 29. **Proposer** L'équation de V_2 en fonction de V_{21}, V_{22}, tws, tws_1 et tws_2. Puis en utilisant V_1 et V_2 **proposer** l'équation de V. **Calculer** le résultat pour un vent de 12 nœuds et un angle de 61°.



Tournez la page S.V.P.

3.2. Utilisation de la polaire de vitesse

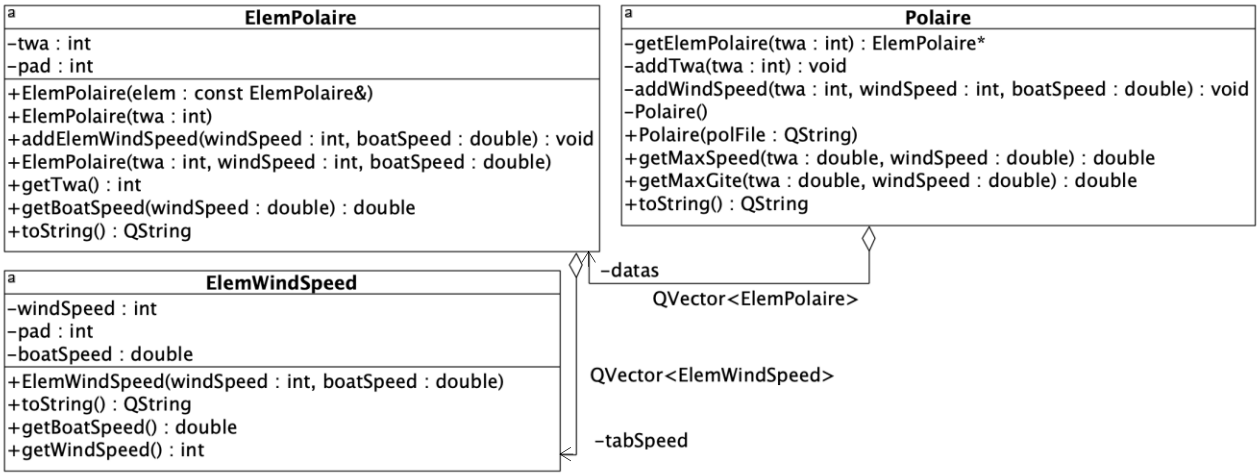


Figure 15 : Prototype de l'architecture objet d'exploitation du fichier polaire

```
class Polaire: public QObject
{
private:
    QVector<ElemPolaire*> datas;
    ElemPolaire* getElemPolaire(int twa);
    void addTwa(int twa);
    void addWindSpeed(int twa, int windSpeed, double boatSpeed);
    Polaire(){}
public:
    virtual ~Polaire(){}
    Polaire(QString polFile); // ouvrir un fichier pol
    double getMaxSpeed(double twa, double windSpeed);
    double getMaxGite(double twa, double windSpeed);
    QString toString();
};
```

La classe polaire a deux rôles :

- Charger les vitesses par extraction des données du fichier texte ascii « .pol » dans des objets contenant les informations TWA, TWS et vitesse du bateau.
- Calculer la vitesse du bateau pour toutes valeurs TWA et TWS par bi-linéarisation.

Q 30. **Proposer** la déclaration en C++ de la classe *ElemPolaire*.

Les méthodes toString() permettent d'afficher le contenu des membres privés des objets (datas inclus).

Q 31. **Proposer** une implémentation en C++ possible de la méthode *toString()* de la classe Polaire.
Hypothèses : la classe QVector a un accesseur par [indice] et la classe QString accepte la concaténation par l'opérateur +.

Le moteur physique ne peut qu'appliquer des forces (N) sur des solides. Proposer une stratégie permettant d'imposer une vitesse de déplacement du navire à partir du moteur physique.

Q 32. **Quelle piste** peut être suivie pour faire se déplacer le navire à la vitesse voulue ? **Conclure** sur cette partie.

Partie 4 : Communication

***Objectif** : valider l'architecture logicielle de RollABoat vis-à-vis de la communication réseau*

4.1. Protocole Modbus

Les échanges entre le poste RollABoat et l'automate sont en ModBus/IP. Le cadencement des échanges est programmé par l'automate à 40ms. La table ci-dessous liste la table d'échange.

Registre	Nb reg	Nom du registre	Description	Type
40001	2	lect-roulis	position roulis (deg), RAB → PO	Float
40003	2	lect-tangage	position tangage (deg), RAB → PO	Float
40005	2	lect-bome	position bome (deg), RAB → PO	Float
40007	2	lect-vitazimut	vitesse azimut (rad/s), RAB → PO	Float
40009	2	reserve	Pour utilisation future, RAB → PO	Float
40011	2	cons-etat	0 = jeu, 1 = manu, 2 = sans PO, PO → RAB	Float
40013	2	cons-gouv	-1/1 position gouvernail, PO → RAB	Float
40015	2	cons-roulis	consigne de position roulis (deg), PO → RAB	Float
40017	2	cons-tangage	consigne de position tangage (deg), PO → RAB	Float
40019	2	cons-difbotwa	diff angle réel bôme réf. bateau et lect-bome, PO → RAB	Float
40021	2	cons-vitazimut	consigne de vitesse azimut (+/-rad/s)	Float
40023	2	cons-acc	consigne de l'accélération (-1 a 1), PO → RAB	Float
40025	2	cons-tws	**T**rue **W**ind **S**peed	Float
40027	2	cons-swa	consigne de la direction du vent **S**ystem **W**ind **A**ngle	Float
40029	2	cons-hautvague	consigne de hauteur de vague (m)	Float
40031	2	cons-vitvague	consigne vitesse de vague (m/s)	Float
40033	2	cons-intervague	consigne de distance inter vagues (m inter crêtes)	Float

Figure 16 : Table d'échange ModBus, chaque Float est stocké sur deux registres de 16 bits.

Q 33. **Quel est** le débit applicatif utile utilisé ? **Quel en est** l'impact vis-à-vis des performances d'un échange WIFI 802.11 b/g/n.

Q 34. Dans le cas où, en exploitation, il y aurait plusieurs simulateurs proches, les postes RollABoat étant reliés en WIFI. **Quelles seraient** les précautions à prendre en compte pour mettre en œuvre les liaisons ?

Soit la requête MODBUS ci-contre (MBAP inclus) : **00 02 00 00 00 06 FF 03 00 00 00 02**

Q 35. **Valider** tous les octets de la trame requête. Voir DT3.

Q 36. **Proposer** une trame de réponse possible pour des données nulles.

Soit l'échange Ethernet ci-dessous (présenté sans les octets de préambule).

Trame 1 :

```
0000   00 d0 c9 17 09 61 00 0f b0 71 c4 c1 08 00 45 00   .....a...q....E.
0010   00 40 00 00 40 00 40 06 00 00 ac 10 26 01 ac 10   .@..@.@.........
0020   26 03 d5 5d 01 f6 58 43 6c f7 c6 eb 5b 9d 80 18   ...]..XCl...[...
0030   18 e9 83 a1 00 00 01 01 08 0a 1e d8 c6 b4 1e d8   ................
0040   c6 19 00 41 00 00 00 06 ff 03 00 0c 00 02         ...A..........
```

Trame 2 :

```
0000   00 0f b0 71 c4 c1 00 d0 c9 17 09 61 08 00 45 00   ...q.......a..E.
0010   00 41 00 00 40 00 40 06 00 00 ac 10 26 03 ac 10   .A..@.@.........
0020   26 01 01 f6 d5 5d c6 eb 5b 9d 58 43 6d 03 80 18   .....]..[.XCm...
0030   18 e9 83 a2 00 00 01 01 08 0a 1e d8 c6 b4 1e d8   ................
0040   c6 b4 00 41 00 00 00 07 ff 03 04 99 9a c0 19      ...A...........
```

Q 37. À partir des trames ci-dessus et de **DT4**, **remplir** le document réponse **DR37**.

4.2. Échange de données

La modélisation objet simplifiée de l'application est présentée ci-dessous.

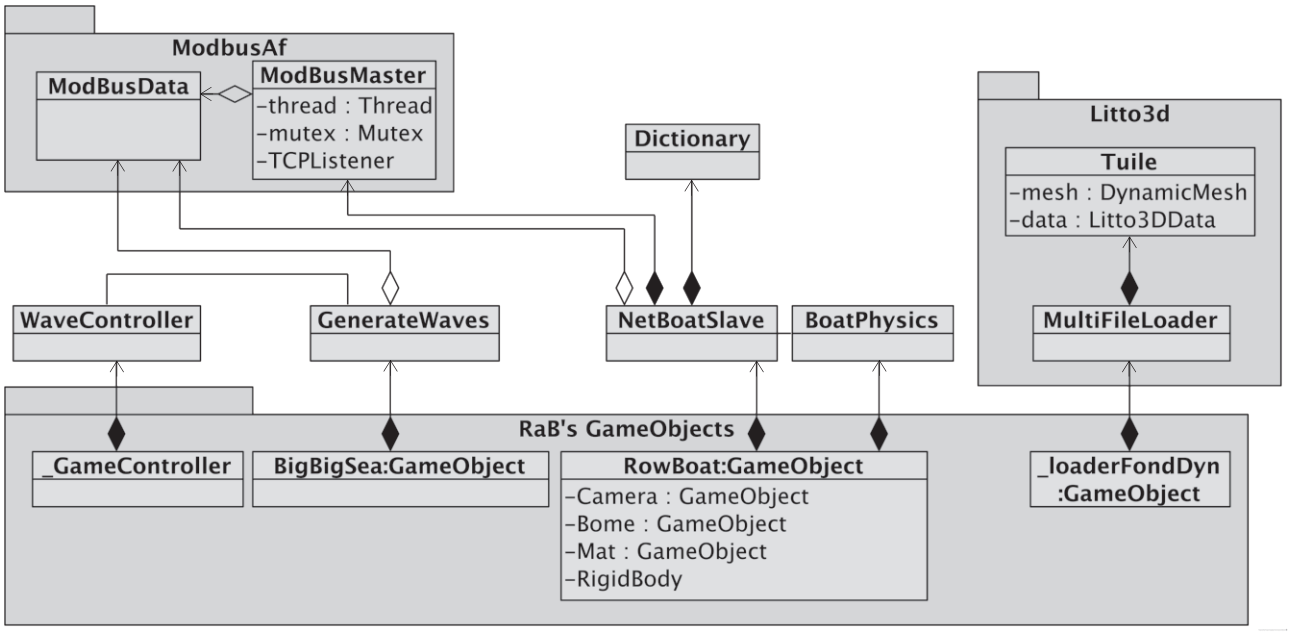


Figure 17 : diagramme de classe/objet de RollABoat

La table des registres est synchronisée entre l'automate et l'application RollABoat suivant le diagramme de séquence suivant :

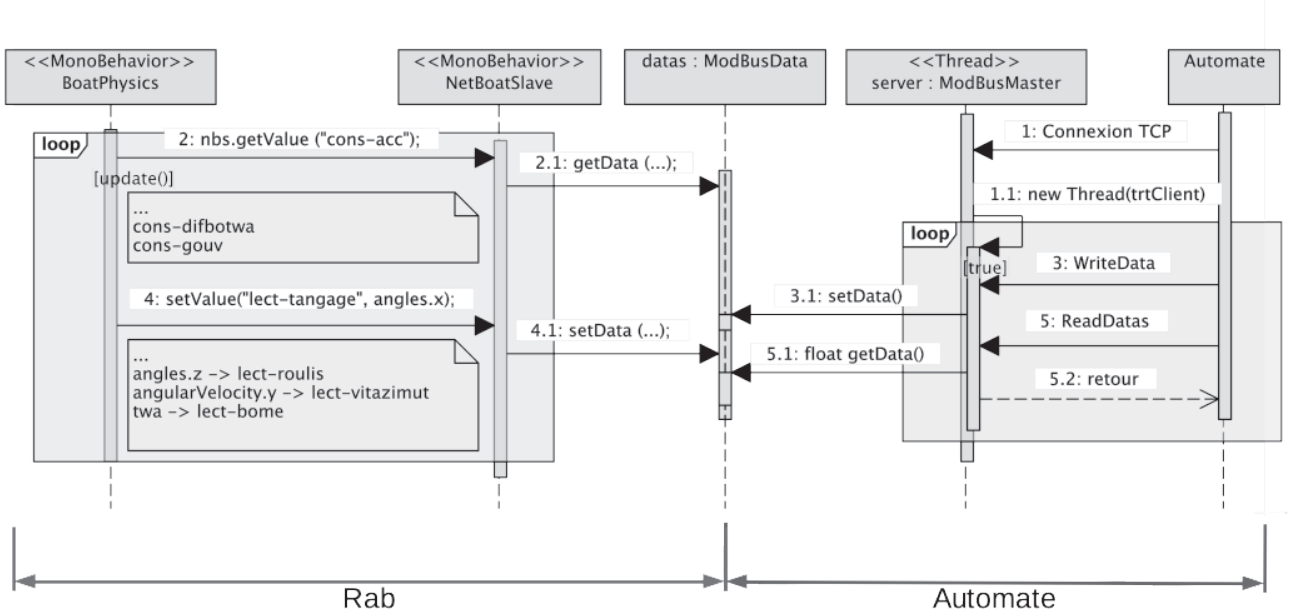


Figure 18 : échange entre la partie simulateur et la partie Modbus de l'application RollABoat

La droite du diagramme nommé « Automate » correspond à l'échange piloté par l'automate. La gauche du diagramme nommé « RaB » (RollABoat) correspond à l'échange piloté par *Unity*. L'échange de données se fait par l'objet ***datas*** de la classe ***ModBusData***. Les données sont stockées sous forme de couple clef (uint32), valeur(float). Ce même couple permet aux méthodes getData(), setData() d'accéder et de modifier les données.

Q 38. À partir des éléments présentés, **proposer** une déclaration de la classe de *ModbusData.*

Dans l'application *ModbusMaster* utilise un thread. Lorsque l'automate viens interroger RollABoat.

Q 39. **Quel est** le rôle de ce thread ?

Q 40. Dans la relation client / serveur, **donner** le rôle joué par l'application *RollABoat.*

Q 41. **Préciser** la relation dans le diagramme de classe entre *NetBoatSlave* et *ModBusMaster*. Comment ce type de relation est implémenté en C++ ou en Python.

Q 42. **Quel mécanisme** doit être mis en œuvre pour protéger *ModbusData* contre les écritures concurrentes ?

Tournez la page S.V.P.

Partie 5 : Mise en exploitation du simulateur

Objectif : *valider l'architecture matérielle et logicielle réalisant l'interface entre la partie opérative et la partie simulation nommée « RollABoat ».*

5.1. Architecture réseau

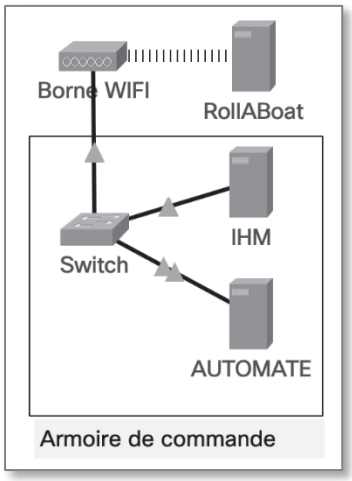


Figure 19 : Architecture réseau prototype

L'architecture réseau utilisée dans ce simulateur prototype est décrite en Figure 19. Elle permet de faire communiquer l'ensemble des équipements.

Dans une réflexion de mise en production de ce simulateur dans une école de voile l'architecture évoluerait de la manière décrite en « Figure 20 ». Les clients internet seraient en mesure de réserver un créneau de simulation sur bateau. Chaque client pourrait choisir son bateau virtuel parmi une liste de bateau disponible et ses paramètres de simulation type de mer, force de vent, zone de navigation. L'ensemble de ces informations est organisé dans une base de données situées sur le « serveur site production ». Le serveur web est stocké dans le « serveur site DMZ ».

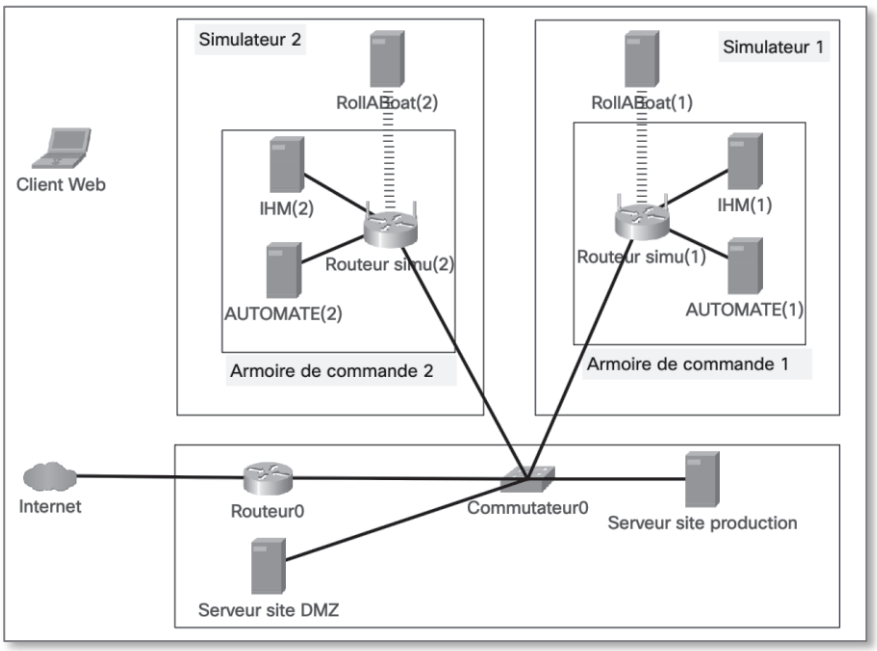


Figure 20 : Architecture réseau production

Le simulateur prototype possède des adresses IP privées dans la plage d'adresse de sous réseaux suivante : 172.16.38.0/28. Cette plage est issue de la plage 172.16.38.0/24. Les « routeurs simu » incluent un commutateur et un point d'accès wifi.
Le routeur0 assure le pare-feu, le routage internet et le routage inter vlan (DMZ inclus). La DMZ est dans un VLAN dédié et routé par *Routeur0*.

Q 43. **Préciser** comment doit être configuré la liaison entre *Routeur0* et *Communateur0* pour qu'une architecture avec un serveur DMZ câblé sur le commutateur soit possible.

Q 44. Dans le document réponse **DR44**, **lister** les adresses IP utilisées Figure 19 par les différents équipements en commençant par le début de la plage.

Q 45. L'architecture réseau en production, Figure 20, possède deux simulateurs. Sur le document réponse **DR45**, **proposer** un plan d'adressage du simulateur 1 et 2 en utilisant les deux premiers sous réseau de la plage d'adresse 172.16.38.0/**24**. Les sous réseaux utilisent le masque /28.

Q 46. Dans le document réponse **DR46**, **Proposer** un plan d'adressage pour les matériels restants et **déterminer** la table de routage de routeur0.
Hypothèse : l'ensemble de la plage 172.16.38.0/24 est disponible.

L'école de voile possède un abonnement professionnel pour l'accès à internet. Cet abonnement comprend un routeur pro nommé **routeur0**, une adresse publique IPv4 ainsi que son nom de domaine « voilesurterre.fr ». Elle a fait inscrire également www sur la même adresse.

voilesurterre.fr	type d'enregistrement :	valeur :	TTL
@	A	11.22.33.44	14400

Q 47. **Quel matériel répond** à un ping sur www.voilesurterre.fr fait sur internet et avec quelle adresse IP ?

Le service web est assuré par le « serveur site DMZ ».

Q 48. **Que faut-il faire** pour que cela soit possible ? Préciser les ports et les matériels impactés par cette configuration.

5.2. Base de données

Le « serveur site production » contient une base de données relationnelle permettant l'exploitation du site et la gestion des clients. Une première réflexion de conception de la base de données conclut à deux types de propositions.
Le premier type de propositions porte sur les entités :

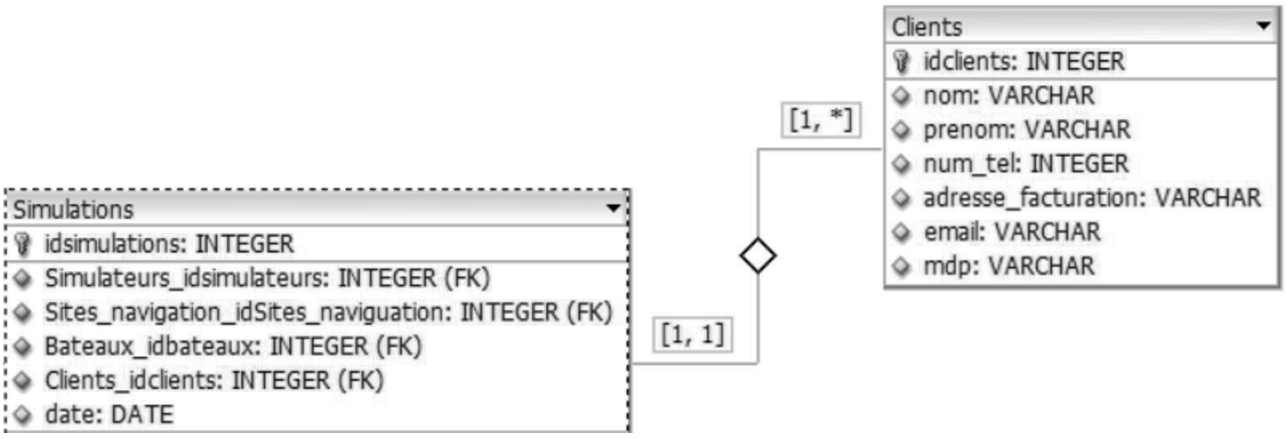
- il existe des clients qui sont caractérisés par les noms, prénoms, n° de client, téléphone, adresse de facturation, @email, mot de passe crypté ;
- il existe des simulateurs caractérisés par l'IP du routeur et le nom du technicien de maintenance ;
- il existe des bateaux à simuler qui sont caractérisés par leur type et leur nom ;
- il existe des sites de navigation caractérisés par des coordonnées GPS, un niveau de difficulté… ;
- il existe des simulations qui sont caractérisées par la date de simulation.

Le deuxième type de proposition mentionne les possibilités du système :

- une simulation porte sur un type de bateau ;
- une simulation utilise un simulateur ;
- un client peut réaliser plusieurs simulations ;
- une simulation porte sur un site de navigation.

Q 49. **Proposer** le schéma conceptuel de la base de données en spécifiant notamment pour chaque entité les clés primaires. **Spécifier** les relations entre chaque entité avec leurs multiplicités.

Pour la suite du sujet la table Clients et la table Simulations seront les suivantes :



Q 50. **Proposer** la requête SQL de création de la table client.

Q 51. **Proposer** la requête SQL listant les noms, prénoms, mdp et email des clients, classé alphabétiquement par nom, puis prénom.

Q 52. **Proposer** la requête SQL permettant d'obtenir la liste des dates des simulations réalisées par un client à partir de son email (exemple : marie.tournelle@llf.fr).

Q 53. **Proposer** une requête SQL permettant d'ajouter une réservation de simulation.

Q 54. **Proposer** la requête SQL permettant d'obtenir le nombre de simulations **par** email client. On ne veut afficher que les clients qui ont fait plus de 3 simulations.

Q 55. On peut avoir besoin de supprimer des clients. **Préciser** la contrainte pour la suppression d'un client de la base de données.

Q 56. Conclure sur la proposition de mise en exploitation de deux simulateurs et des impacts de celle-ci.

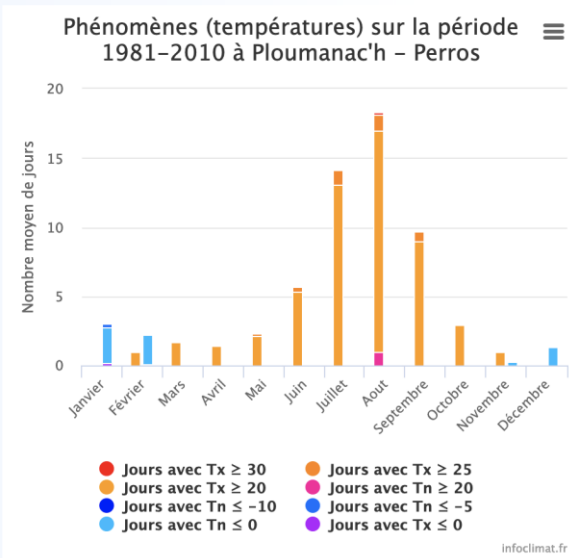
DOCUMENTS TECHNIQUES

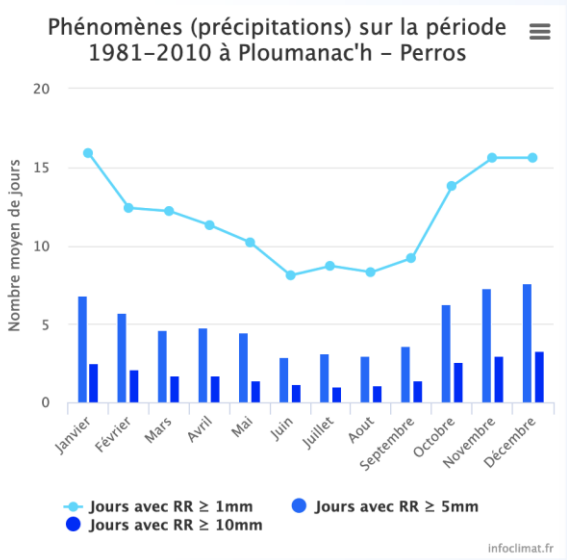
DT1 – Statistiques station météo

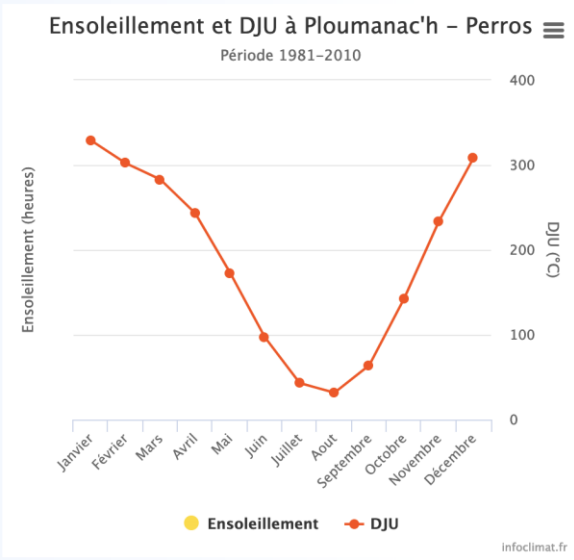
Station de référence :

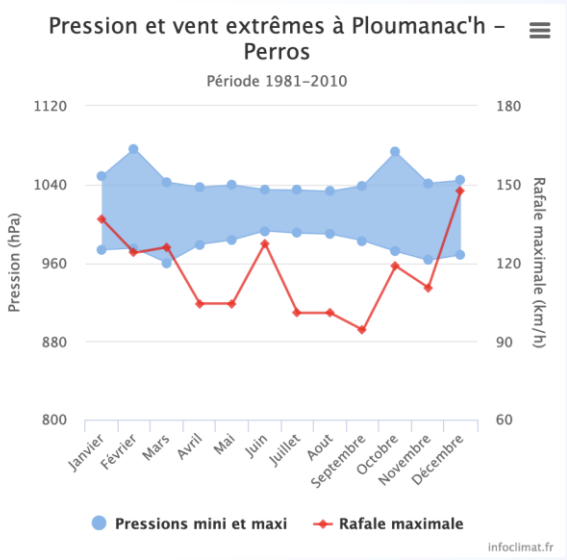
Ploumanac'h - Perros

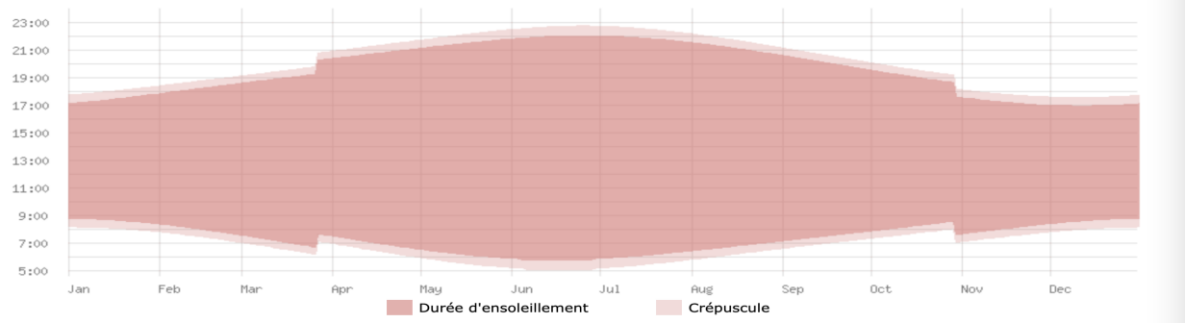
	janv.	fev.	mars	avr.	mai	juin	juil.	août	sept.	oct.	nov.	dec.	Toute la période
Vent ≥ 57.6 (km/h- jours)	17.2	13.8	13.3		6.9	3.9	5.1	3.9	6		13.5	17.2	100.8
Vent ≥ 100.8 (km/h- jours)	1.6	0.5	0.2		0.3	0.1	0.1	0.1			0.2	0.9	4











DT2 – Documentation Vector3 Unity 3D

La classe Vector3 est utilisée pour représenter un vecteur tridimensionnel dans l'espace ou un point.

Constructeurs

- **Vector3(float x, float y, float z)** : Crée un nouveau vecteur avec les composantes spécifiées.
- **Vector3(float value)** : Crée un nouveau vecteur avec toutes les composantes égales à la valeur spécifiée.
- **Vector3(Vector2 vector, float z)** : Crée un nouveau vecteur en utilisant les composantes x et y du vecteur spécifié et la valeur spécifiée pour la composante z.

Propriétés

- **magnitude** (type : float) : La magnitude (longueur) du vecteur.
- **normalized** (type : Vector3) : Le vecteur normalisé (de magnitude 1).
- **sqrMagnitude** (type : float) : La magnitude au carré du vecteur.

Méthodes

- **Cross(Vector3 vector)** (type : Vector3) : Calcule le produit vectoriel entre le vecteur et le vecteur spécifié.
- **Distance(Vector3 vector)** (type : float) : Calcule la distance entre le vecteur et le vecteur spécifié.
- **Dot(Vector3 vector)** (type : float) : Calcule le produit scalaire entre le vecteur et le vecteur spécifié.
- **Lerp(Vector3 to, float t)** (type : Vector3) : Interpole linéairement entre le vecteur et le vecteur spécifié en utilisant le facteur spécifié t (0 = vecteur, 1 = vecteur spécifié).
- **Max(Vector3 vector)** (type : Vector3) : Retourne un vecteur contenant la plus grande valeur de chaque composante entre le vecteur et le vecteur spécifié.
- **Min(Vector3 vector)** (type : Vector3) : Retourne un vecteur contenant la plus petite valeur de chaque composante entre le vecteur et le vecteur spécifié.
- **MoveTowards(Vector3 to, float maxDistanceDelta)** (type : Vector3) : Déplace le vecteur vers le vecteur spécifié avec une distance maximale spécifiée.
- **Normalize()** (type : void) : Normalise le vecteur (magnitude = 1).
- **Reflect(Vector3 normal)** (type : Vector3) : Réfléchit le vecteur autour du vecteur spécifié.
- **Scale(Vector3 scale)** (type : void) : Multiplie chaque composante du vecteur par la composante correspondante du vecteur spécifié.
- **ToString()** (type : string) : Convertit le vecteur en une chaîne de caractères.
- **Set(float x, float y, float z)** (type : void) : Définit les valeurs des composantes du vecteur.
- **Slerp(Vector3 to, float t)** (type : Vector3) : Interpole sphériquement entre le vecteur et le vecteur spécifié en utilisant le facteur spécifié t (0 = vecteur, 1 = vecteur spécifié).
- **SmoothDamp(Vector3 target, ref Vector3 currentVelocity, float smoothTime, float maxSpeed)** (type : Vector3) : Calcule une position douce et amortie entre le vecteur et le vecteur spécifié.

Chaque méthode retourne une valeur de type spécifique, comme indiqué ci-dessus.

Opérateurs de surcharge

- **+ (addition) :** Ajoute deux vecteurs et retourne un nouveau vecteur résultant de l'addition.
- **- (soustraction) :** Soustrait un vecteur d'un autre vecteur et retourne un nouveau vecteur résultant de la soustraction.

Tournez la page S.V.P.

- *** (multiplication) :** Multiplie un vecteur par un scalaire (float) et retourne un nouveau vecteur résultant de la multiplication.
- **/ (division) :** Divise un vecteur par un scalaire (float) et retourne un nouveau vecteur résultant de la division.
- **== (égalité) :** Compare si deux vecteurs sont égaux en termes de valeurs de leurs composantes et retourne un booléen indiquant si les vecteurs sont égaux ou non.
- **!= (différence) :** Compare si deux vecteurs sont différents en termes de valeurs de leurs composantes et retourne un booléen indiquant si les vecteurs sont différents ou non.
- **- (négation) :** Négation un vecteur et retourne un nouveau vecteur dont les composantes sont les opposées du vecteur d'origine.
- **[] (indexeur) :** Permet d'accéder aux composantes d'un vecteur en utilisant des indices (0 pour x, 1 pour y, 2 pour z). et constructeur

Exemples

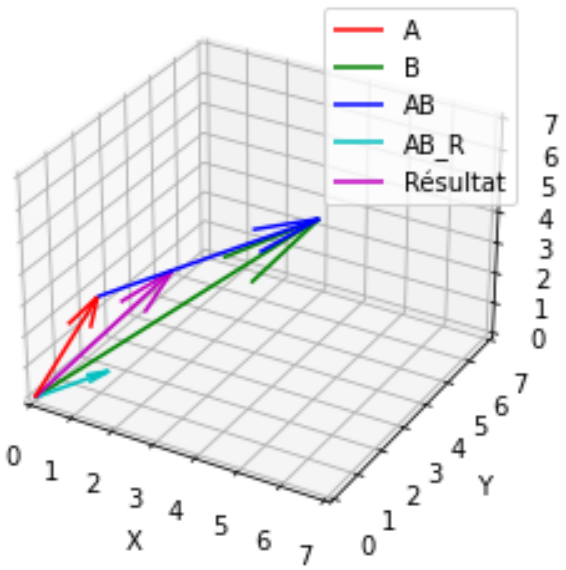
```
// Déclaration des points A et B
Vector3 pointA = new Vector3(1.0f, 1.0f, 3.0f);
Vector3 pointB = new Vector3(4.0f, 5.0f, 4.0f);

// Création du vecteur AB en soustrayant le point A du point B
Vector3 vecteurAB = pointB - pointA;

Debug.Log("Vecteur AB : " + vecteurAB);
// Vecteur AB : (3.0, 4.0, 1.0)

// Multiplication du vecteur AB par 1/3
vecteurAB_R = AB * (1.0f / 3.0f);

// Addition du résultat avec le point A
Vector3 resultat = vecteurAB_R + pointA;
```



DT3 – Modbus over TCP/IP

Modbus is an open protocol, meaning that it's free for manufacturers to build into their equipment without having to pay royalties. It has become a standard communications protocol in industry, and is now the most commonly available means of connecting industrial electronic devices. It is used widely by many manufacturers throughout many industries.

Data tables

Information is stored in the Slave device in four different tables. Two tables store on/off discrete values (coils) and two store numerical values (registers). The coils and registers each have a read-only table and read-write table. Each table has 9999 values. Each coil or contact is 1 bit and assigned a data address between 0000 and 270E. Coil/Register Numbers can be thought of as location names since they do not appear in the actual messages. The Data Addresses are used in the messages. For example, the first Holding Register, number 40001, has the Data Address 0000. The difference between these two values is the offset. Each table has a different offset: 1, 10001, 30001 and 40001.

Coil/Register Numbers	*Data Addresses*	*Type*	*Table Name*
1-9999	0000 to 270E	Read-Write	Discrete Output Coils
10001-19999	0000 to 270E	Read-Only	Discrete Input Contacts
30001-39999	0000 to 270E	Read-Only	Analog Input Registers
40001-49999	0000 to 270E	Read-Write	Analog Output Holding Registers

Function codes

This number tells the slave which table to access and whether to read from or write to the table.

Function Code	*Action*	*Table Name*
1 (01 hex)	Read	Discrete Output Coils
5 (05 hex)	Write single	Discrete Output Coil
15 (0F hex)	Write multiple	Discrete Output Coils
2 (02 hex)	Read	Discrete Input Contacts
4 (04 hex)	Read	Analog Input Registers
3 (03 hex)	Read	Analog Output Holding Registers
6 (06 hex)	Write single	Analog Output Holding Register
16 (10 hex)	Write multiple	Analog Output Holding Registers

Modbus over TCP/IP format = MBAP header + PDU (Protocol Data Unit)

MBAP Header

Over IP, a 7-byte header called the MBAP header (Modbus Application Header) is added to the start of the message. This header has the following data:

- Transaction Identifier: 2 bytes set by the Client to uniquely identify each request. These bytes are echoed by the Server since its responses may not be received in the same order as the requests.
- Protocol Identifier: 2 bytes set by the Client, always = 00 00
- Length: 2 bytes identifying the number of bytes in the message to follow.
- Unit Identifier: 1 byte set by the Client and echoed by the Server for identification of a remote slave connected on a serial line or on other buses.

Tournez la page S.V.P.

PDU Transaction Formats (registers function code only)

Read Holding Registers (FC=03)

request

0	1	2	3	4
03	address		number	

number = 1…125 (007D) , N = number * 2

response

0	1	2	3	…
03	N	data (N bytes)		

Read Input Registers (FC=04)

request

0	1	2	3	4
04	address		number	

number = 1…125 (007D) , N = number * 2

response

0	1	2	3	…
04	N	data (N bytes)		

Preset Single Register (FC=06)

request

0	1	2	3	4
06	address		value	

value = 0…65535 (FFFF)

response = echo of the request

Preset Multiple Registers (FC=16)

request

0	1	2	3	4	5	6	…
10	address		number		N	data	

number = 1…120 (0078) , N = number * 2

response

0	1	2	3	4
10	address		number	

Exception Responses

Following a request, there are 4 possible outcomes from the slave.

- The request is successfully processed by the slave and a valid response is sent.
- The request is not received by the slave therefore no response is sent.
- The request is received by the slave with a parity, CRC or LRC error. The slave ignores the request and sends no response.
- The request is received without an error, but cannot be processed by the slave for another reason. The slave replies with an exception response.

In a normal response, the slave echoes the function code. The first sign of an exception response is that the function code is shown in the echo with its highest bit set. All function codes have 0 for their most significant bit. Therefore, setting this bit to 1 is the signal that the slave cannot process the request. In this case, following the Function Code is the Exception Code. The exception code gives an indication of the nature of the problem.

DT4 – Protocoles usuels

Ethernet

Octet 1	Octet 2	Octet 3	Octet 4
Préambule			
préambule			
adresse destination (0-3)			
adresse destination (4-5)		adresse source (0-1)	
adresse source (2-5)			
type des données		données (0 à 1500 octets)	
bourrage (0 à 46 octets)			
contrôle			

préambule : 7 octets égaux à 10101010 + 1 octet égal à 10101011
adresse destination : codée sur 48 bits (FF:FF:FF:FF:FF:FF pour diffusion)
adresse source : codée sur 48 bits
type des données : identification du protocole couche supérieure
(0x0800 pour IP, 0x0806 pour ARP ...)
données : données de la couche supérieure
bourrage : la longueur minimale d'une trame ethernet est de 64 octets (sans compter le préambule). Si les données ont une longueur inférieure à 46 octets, le champ bourrage est utlisé.

IP

	Octet 1		**Octet 2**	**Octet 3**		**Octet 4**
Mot 1	Version	Long	Type de service	Longueur totale		
Mot 2	Identification			Flags	Position fragment	
Mot 3	Temps de vie		Protocole	Checksum en-tête		
Mot 4	Adresse station source					
Mot 5	Adresse station destinataire					
Mot 6	Options					Bourrage

LONG : longueur de l'en-tête IP comptée en mots de 32 bits.
Type de service : désigne la qualité de service désirée pour le datagramme.

Précédence	D	T	R	0	0

D signifie Delay
T signifie Troughput (débit)
R signifie Reliability (fiabilité)
Longueur totale : longueur totale du datagramme mesurée en octets, y compris l'en-tête IP et les données.
Identification : identification pour reconstituer les différents fragments d'un datagramme.
Flags : champ de trois bits gérant la fragmentation
000 – autorise la fragmentation, dernier fragment,
001 – autorise la fragmentation, ce n'est pas le dernier fragment,
010 – fragmentation non autorisée.
Position : indique la position d'un fragment, comptée en octet par rapport au début des données du paquet complet. Si le datagramme est complet, ou s'il s'agit du premier fragment, ce champ est à 0.

Tournez la page S.V.P.

Temps de vie : indique la durée de vie maximale du datagramme au travers du réseau. La valeur par défaut est de 15 secondes. Le paquet est détruit si le temps de vie est dépassé.
Protocole : définit le numéro de SAP qui recevra le paquet.

liste des protocoles les plus connu :
- 01 - 00001 - ICMP
- 02 - 00010 - IGMP
- 06 - 00110 - TCP
- 17 - 10001 - UDP

Checksum de l'en-tête : somme complémentée à 1 de l'en-tête.
Adresse source et destination : adresses codées sur 32 bits.
Options : champ de longueur variable en fonction du type et du nombre d'options.
Bourrage : valeur de remplissage pour obtenir une en-tête avec un nombre entier de mots de 32 bits.

TCP

	Octet 1	**Octet 2**	**Octet 3**	**Octet 4**
Mot 1	Port source		Port destination	
Mot 2	Numéro de séquence			
Mot 3	Numéro d'acquittement			
Mot 4	Contrôle		Fenêtre	
Mot 5	Checksum		Urgent pointer	
Mot 6	Options			
Mot 7	Donnnées			

- **Port source et destination :** nombres de 16 bits qui identifient la connexion (correspondent à l'application de la couche supérieure,
- **Numéro de séquence :** permet de rétablir l'ordre des paquets reçus et d'éliminer les paquets dupliqués,
- **Numéro d'acquittement :** si le flag ACK est présent, désigne le prochain numéro de séquence qui sera transmis par l'autre bout de la connexion,
- **Contrôle** :
  offset données (4 bits) : indique le nombre de mots de 32 bits de l'en-tête TCP,
  réservé (6 bits) : zone toujours à zéro,
  FLAGS (6 bits) : zone composée des bits U, A, P, R, S, F
  U : segment à traiter en Urgence,
  A : le segment transporte un numéro d'acquittement significatif,
  P : si le paquet reçu comporte ce flag, TCP le transmet immédiatement à la couche supérieure sans attendre d'autres segments (utile par exemple pour un fonctionnement correct d'un écho sur une console),
  R : provoque un Reset de la connexion,
  S : demande de connexion, permet la synchronisation des numéros de séquence,
  F : Fin, plus de données à transmettre.
- **Fenêtre** : indique le nombre d'octets qui seront acceptés par la station émettrice à partir de celui indiqué par le numéro d'acquittement présent dans le segment.
- **Checksum** : complément à 1 de la somme des mots de 16 bits de l'en-tête TCP et des données,
- **Urgent pointer :** si le flag U est à 1, indique en nombre d'octets le décalage par rapport au numéro de séquence de ce segment des données à traiter en urgence,
- **Options** :

type	longueur	signification
0	-	fin de la liste des options
1	-	pas d'opération
2	4	longueur maximum d'un segment
4	-	SACK permis

- **Données** : données provenant de la couche supérieure à TCP.

Modèle CMEN v3

Nom de famille :
(Suivi, s'il y a lieu, du nom d'usage)

Prénom(s) :

Numéro Candidat :

Né(e) le : / /

Cadre réservé aux candidats de concours de recrutement et examens professionnels

Concours : ..... **Option / Section :** ..... **N° d'inscription :**

Cocher une seule case parmi les six types de concours suivants :

☐ externe ☐ 3e externe ☐ externe spécial ☐ interne ou 1er interne ☐ 2nd interne ☐ 2nd interne spécial

Cocher public OU privé UNIQUEMENT pour les concours enseignants : ☐ public ☐ privé

Examen professionnel pour l'avancement au grade de : .....

Cadre réservé aux candidats d'examens et du concours général

Examen : ..... **Série / Spécialité :** .....

Epreuve - Matière : ..... **Session :** .....

EDE NUM 1

DR11 - DR37 - DR44 - DR45 - DR46

Tous les documents réponses sont à rendre, même non complétés.

NE RIEN ECRIRE DANS CE CADRE

Document réponse **DR46**
Proposer la suite du plan d'adressage

	Équipements	Adresse IP (à compléter)
Sous réseau N°3	Serveur site production	172 . . . /
	Routeur0-production	172 . . . /
	Routeur simu 1 Côté Routeur0	172 . . . /
	Routeur simu 2 Côté Routeur0	172 . . . /
Sous réseau N°4	Serveur site DMZ	172 . . . /
	Routeur0-DMZ :	172 . . . /

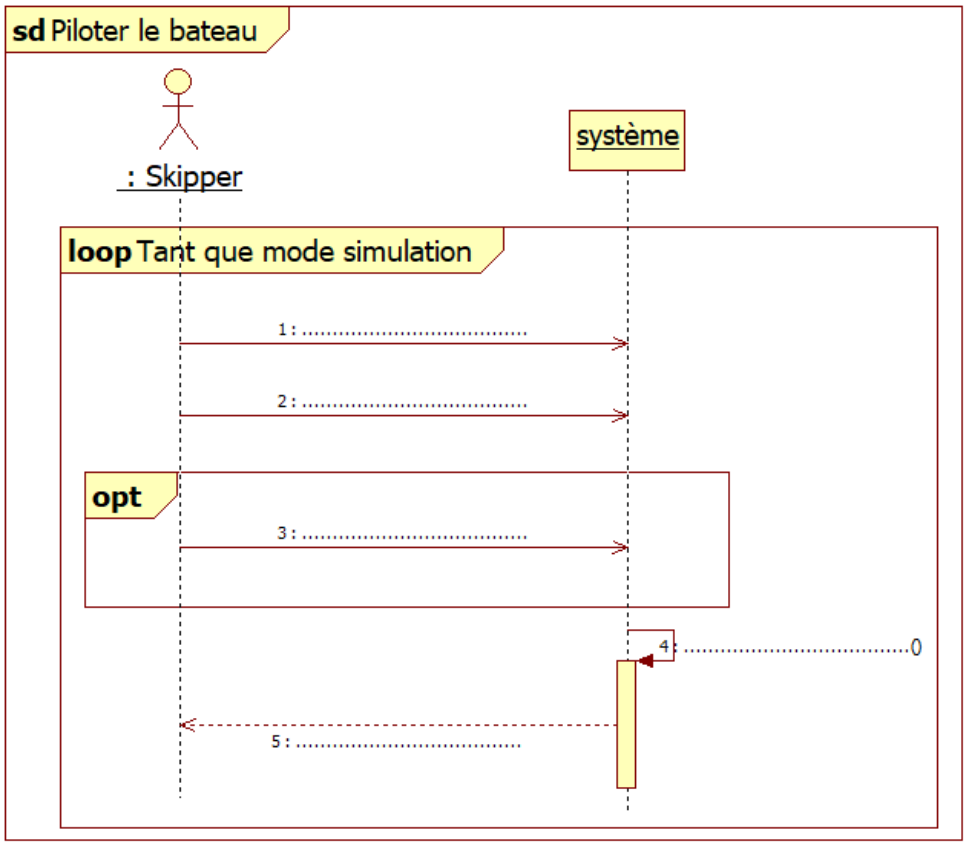
Table de routage de routeur0.

(Rq : le nombre de lignes ne présume pas un nombre de routes statiques)

Réseau	Masque	IP passerelle	Interface de sortie

DOCUMENTS RÉPONSES

Document réponse **DR11**



Document réponse **DR37**

Trame N°1 :

Type de trame	
IP émetteur	
IP récepteur	
Port destination	
Registre(s) concerné(s)	
Données brutes	
Données décodées	
Nom du registre / Nom des registres	

Trame N°2 :

Type de trame	
IP émetteur	
IP récepteur	
Port destination	
Registre(s) concernés	
Données brutes	
Données décodées	
Nom du registre / Nom des registres	

Tournez la page S.V.P.

Document réponse **DR44**

Équipements (à compléter)	Adresse IP (à compléter)
AUTOMATE	172 . . . /

Document réponse **DR45**

	Équipements simulateur 1	Adresse IP (à compléter)
Sous réseau N°1	AUTOMATE(1)	172 . . . /
	IHM (1)	172 . . . /
	RollABoat (1)	172 . . .
	Routeur simu1 côté armoire de commande	172 . . . /

	Équipements simulateur 2	Adresse IP (à compléter)
Sous réseau N°2	AUTOMATE (2)	172 . . . /
	IHM (2)	172 . . . /
	RollABoat (2)	172 . . . /
	Routeur simu2 côté armoire de commande	172 . . . /