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

EDE NUM 2

SESSION 2021

CAPET
CONCOURS EXTERNE
ET CAFEP CORRESPONDANT

Section : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR

Option : INGÉNIERIE INFORMATIQUE

ÉTUDE D'UN SYSTÈME, D'UN PROCÉDÉ OU D'UNE ORGANISATION

Durée : 4 heures

Calculatrice électronique de poche - y compris calculatrice programmable, alphanumérique ou à écran graphique – à fonctionnement autonome, non imprimante, autorisée conformément à la circulaire n° 99-186 du 16 novembre 1999.

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

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.

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	1413E	102	7048

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

Concours	Section/option	Epreuve	Matière
EDF	1413E	102	7048

Ce sujet comporte :

- Le contexte de l'étude de la page 2 à 3;
- le travail demandé de la page 3 à 20 ;
- les documents techniques (DT) de la page 21 à 30;
- les documents réponses (DR) de la page 31 à 36 (les DR portent les numéros des questions auxquelles ils se rapportent).

IMPRIMERIE NATIONALE – 21 0224 – D'après documents fournis

SIMULATEUR D'INTERVENTIONS HÉLIPORTÉES

Contexte de l'étude

En France, la flotte aérienne de la sécurité civile et les moyens de maintenance et de soutien sont rassemblés dans le groupement hélicoptères (GH), entité faisant partie intégrante, au sein du GMA (Groupement des moyens aériens), de l'administration centrale du ministère de l'intérieur. Les moyens opérationnels sont à disposition des préfets de département pour emploi. Dédiée aux opérations de secours aux personnes, la flotte d'hélicoptères peut également recevoir des missions d'assistance technique au profit des administrations, ou dans le cadre de la lutte contre les feux de forêts ou certaines pollutions, ainsi que des missions de police.

Assez homogène et largement modernisée, la flotte est composée essentiellement d'une trentaine d'appareils EC 145 acquis auprès d'Eurocopter Deutschland sur un marché commun avec la DGGN (Direction générale de la gendarmerie nationale).

L'EC 145 est un appareil bimoteur, plus rapide et plus puissant que les Alouette III et Dauphin qu'il a remplacés.
Il peut emporter jusqu'à huit passagers ou 500 kg de fret, avec deux membres d'équipage.

L'EC 145 peut franchir 500 km, à une vitesse maximale de 240 km/h.

Figure 1 : EC 145 Sécurité Civile

La flotte d'hélicoptères est répartie sur le territoire métropolitain entre 21 bases, et une base outre-mer (Guadeloupe). L'activité aérienne se situe autour de 18 000 heures par an pour l'ensemble du parc.

En Île-de-France, un appareil EC 145 jaune et rouge de la sécurité civile, jusqu'à présent basé à Issy-les-Moulineaux (Hauts-de-Seine, 91), a été récemment affecté à l'aérodrome de Melun-Villaroche (Seine-et-Marne, 77), où se relaient des équipes des Samu 77 et 91 et des pompiers franciliens.
Melun-Villaroche est un point d'ancrage pour intervenir en Seine-et-Marne et en Essonne, mais aussi dans un rayon de 110 km, jusqu'au nord de l'Yonne par exemple ; la machine couvre ainsi 3 430 communes et 16 millions d'habitants.

En pratique, le taux de charge 24h/24 des appareils (ou taux d'utilisation), toutes missions confondues et en tenant compte des opérations de maintenance, est de l'ordre de 95 % ; ils sont de fait difficilement utilisables pour des actions de formation du personnel.

Il y a quelques années, le SDIS 77 (Service départemental d'incendie et de secours de Seine-et-Marne) a donc souhaité développer un outil de formation aux interventions héliportées qui soit indépendant des appareils en service.

Figure 2 : Alouette III Sécurité Civile

Le système est développé à partir d'une cellule réelle d'aéronef réformé de type Alouette III (machine fournie par l'UDSP77, Union Départementale des Sapeurs-Pompiers de Seine-et-Marne) ; il est destiné à former aussi bien le personnel médical embarqué que le personnel de guidage au sol.

Pour répondre aux besoins, le système doit permettre de simuler des situations de vol et de jouer des scénarios d'interventions contrôlés par un formateur extérieur (intervention sur incendie, sur accident de la route…).

Le projet est nommé « 1790 », en hommage au numéro de série de la cellule d'origine.

La partie du projet relative aux scénarios n'est cependant pas étudiée ici.
La présente étude concerne dans un premier temps les aspects contextuels du système ; elle se poursuit par l'analyse de certaines des solutions techniques mises en œuvre pour exploiter la cellule d'hélicoptère afin de réaliser le simulateur d'interventions. La dernière partie concerne l'intégration du système dans le centre de formation du SDIS.

Les points suivants vont être abordés :

1. Impact socio-économique lié à l'emploi de l'hélicoptère pour le secours à personnes ;
2. Instrumentation des commandes de vol de la cellule ;
3. Conception du système de projection 3D ;
4. Reconstitution du tableau de bord (instruments de vol) ;
5. Architecture réseau du système.

Partie 1 : Impact socio-économique

Objectifs : Appréhender l'importance des moyens héliportés dans le cadre du secours à victimes, et l'obligation de formation professionnelle nécessaire à la mise en œuvre de ces moyens.

Le SDIS 77 et la Sécurité Civile sont notamment amenés à intervenir sur les accidents de la route ; soit par exemple un accident au carrefour de l'Obélisque à Fontainebleau nécessitant l'évacuation en urgence d'un blessé grave vers l'hôpital de la Pitié-Salpêtrière à Paris.
La solution appliquée consiste en général à utiliser un VSAV (véhicule de secours et d'assistance aux victimes) pour parcourir les 65 km avec une vitesse moyenne de 60 km/h pouvant chuter à moins de 45 km/h suivant les conditions de circulation, même avec un véhicule prioritaire.
Avec un hélicoptère EC145, il faut pour le même trajet compter 56 km à vol d'oiseau, à la vitesse moyenne de 4 km par minute, et ajouter dix minutes pour l'ensemble des phases de décollage et d'atterrissage.

Q1 - **Calculer** le bénéfice minimal (en minutes) apporté par la solution héliportée sur le temps d'évacuation Fontainebleau → Paris.

Q2 - **Estimer** le coût pour l'évacuation en VSAV à raison d'environ 1 000 € la demi-heure ; sans tenir compte du retour du véhicule en Seine-et-Marne.

Pour utiliser un EC145, il faut compter 3 600 € l'heure de vol, et ajouter un forfait décollage de l'ordre de 2 000 €.

Q3 - **Calculer** le coût de l'intervention en EC145 en comptant une demi-heure préliminaire entre un démarrage à froid de la base Melun et l'arrivée sur site à Fontainebleau ; mais sans tenir compte du retour de la machine à sa base (le départ de Paris pouvant faire partie d'une autre mission…).

Les principaux personnels concernés par l'utilisation de l'hélicoptère sont les navigants (pilotes et personnel médical) et les professionnels assurant le guidage au sol sur zone. Ces personnes doivent être formées et entrainées pour être efficaces le moment venu. La solution idéale (mais extrêmement onéreuse) consisterait évidemment à disposer d'une flotte réelle d'aéronefs pour cet usage…

Q4 - **Calculer**, à partir du taux de charge donné précédemment, la disponibilité théorique hebdomadaire (en heures entières) d'un appareil réel.

Le programme de formation théorique et pratique est conçu pour des contingents de 12 personnes, avec 32 heures de travaux pratiques par groupes de 3 personnes.

Q5 - Sans le simulateur, **calculer** le nombre de séances requises pour la formation pratique d'un contingent en fonction de la disponibilité d'un appareil réel (en supposant possible de ne réserver l'appareil qu'une seule fois par semaine et pendant seulement la moitié de sa disponibilité théorique maximale).

Q6 - **Conclure** quant à la pertinence de l'utilisation de l'hélicoptère pour l'évacuation de blessés graves.

Q7 - **Conclure** quant à l'intérêt du simulateur d'interventions.

Partie 2 : Instrumentation des commandes de vol

Objectif : Étudier les solutions retenues pour rendre exploitables logiciellement les commandes mécaniques de vol d'un hélicoptère d'ancienne génération.

2.1. Commandes d'une voilure tournante

Un hélicoptère est un aéronef à voilure tournante dont la propulsion et la sustentation sont assurées seulement par des rotors, durant toutes les phases de vol. Cet appareil peut effectuer des manœuvres qu'un avion ne peut faire : vol stationnaire, décollage et atterrissage à la verticale ; à basse altitude, il peut aussi avancer, reculer ou se déplacer latéralement ce qui lui permet d'atteindre des endroits inaccessibles pour un avion.
L'hélicoptère est cependant de conception plus complexe qu'un avion et représente un système fortement instable couplé avec un temps de réponse relativement long ; les machines modernes telles que l'EC 145 disposent d'un calculateur autopilote capable de stabiliser et de contrôler tous les mouvements de l'hélicoptère.

Un hélicoptère dispose potentiellement des 6 degrés de liberté dans l'espace ; les déplacements Tx, Ty, Tz dans les 3 dimensions (avec 0X axe d'avancement) et les rotations Rx, Ry, Rz autour des 3 axes : le roulis (*roll*), le tangage (*pitch*) et le lacet (*yaw*).

Les commandes de vol sont au nombre de quatre (cf. document technique DT 1) ; chacune des commandes est matérialisée par un actionneur fournissant une information physique continue dans une plage limitée mécaniquement.

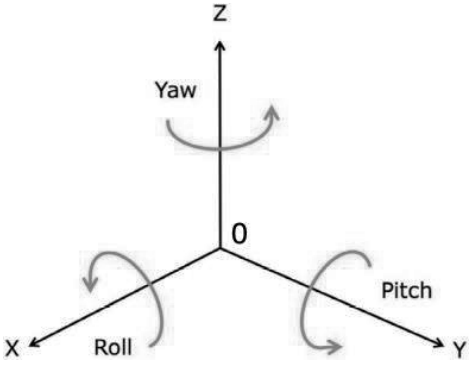


Figure 3 : Degrés de liberté

Tournez la page S.V.P.

Q8 - **Quel est le rôle** du rotor de queue d'un hélicoptère ?

Q9 - **Indiquer** par des croix, pour chaque degré de liberté, la principale commande de vol qui lui est associée (document réponse DR 9).

Le simulateur réalisé est basé sur une cellule d'Alouette III réformée. Cette machine mythique est d'ancienne génération, sans équipement mécatronique, ses commandes de vol sont basées sur des transmissions mécaniques par bielles. Il a donc été nécessaire d'instrumenter ces mécanismes afin de récupérer des informations électriques proportionnelles aux actions du pilote.

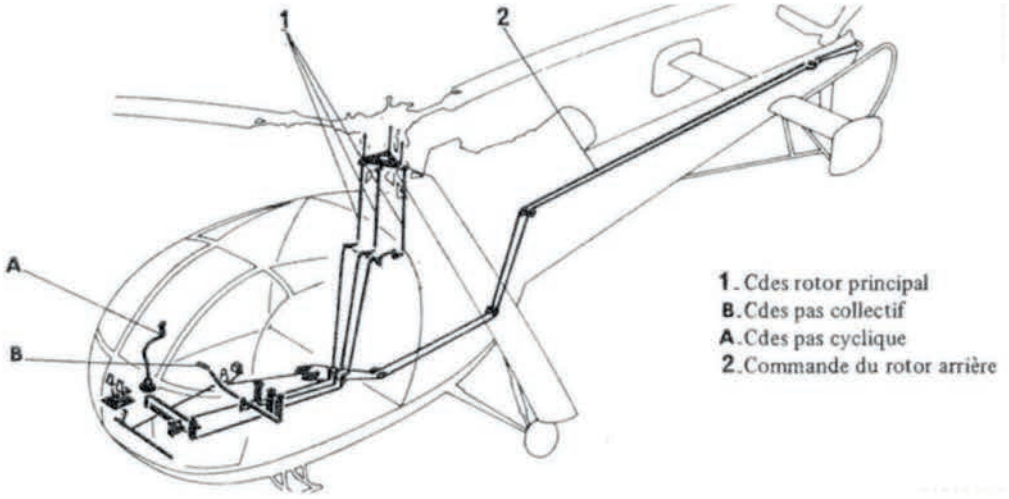


Figure 4 : Commandes de vol de l'Alouette III

2.2. Mise en place des capteurs

La mécanique de renvoi sous plancher étant disponible, la solution économique retenue consiste à mettre en place, pour chaque commande de vol, un potentiomètre rotatif monotour (cf. DT 2) couplé à une poulie sur laquelle un câble souple (type corde acier de guitare) fait un tour mort ; le câble est lui-même fixé aux extrémités sur une bielle d'origine.

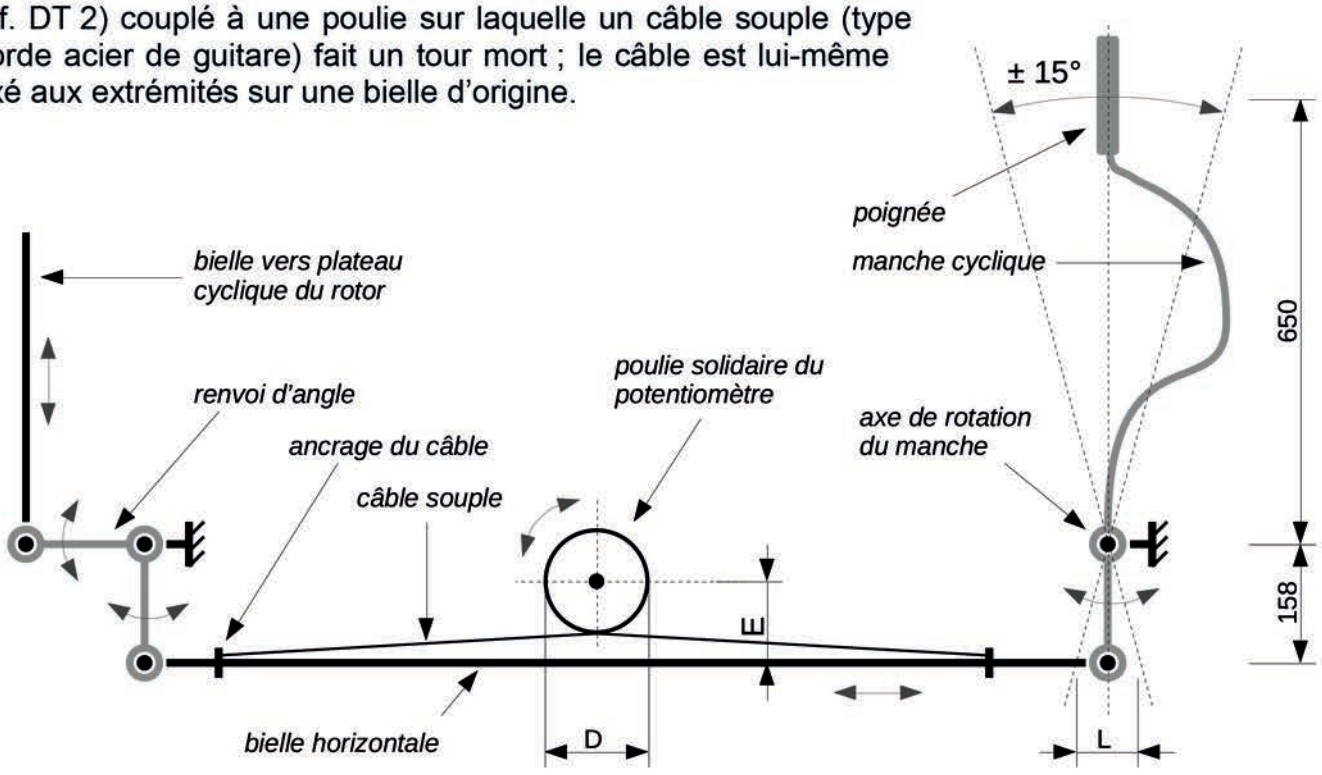


Figure 5 : Schéma de principe pour l'axe longitudinal 0X du manche (dimensions en mm)

Q10 - **Exprimer** de manière littérale le diamètre D (en mm) de la poulie en fonction du débattement horizontal L de la bielle, avec une course du potentiomètre limitée à 90 % de la course nominale (cf. DT 2).

Q11 - **Calculer** le débattement L de la bielle (mm) pour l'axe longitudinal 0X (figure 5).

Q12 - **En déduire** le diamètre minimal de la poulie (mm).

La manipulation du manche induit un léger battement vertical de la bielle horizontale.

Q13 - **À quelle distance** E minimale de la bielle doit être positionné l'axe de la poulie (lorsque la commande est en position centrale comme sur la figure 5) ?

2.3. Acquisition des grandeurs physiques

Le simulateur est couplé au logiciel professionnel X-Plane utilisé notamment pour la formation au pilotage dans l'aviation civile.
Les simulateurs de vol acceptent traditionnellement différents types de manettes de jeux multi-axes en tant qu'actuateurs des commandes de vol des aéronefs ; X-Plane ne faisant pas exception, la solution retenue consiste à transformer les informations des capteurs de manière à ce qu'elles soient reconnues comme issues d'un joystick USB standard compatible HID (*Human Interface Device*).
HID, ou périphérique d'interface humaine, est un appareil informatique qui interagit directement avec les humains. Il est capable de recevoir des données par le biais d'un périphérique d'entrée et renvoie en retour les informations reçues via un périphérique de sortie (le terme HID fait le plus souvent référence à la spécification USB-HID, mais d'autres protocoles, tels que le Bluetooth peuvent aussi exploiter le profil HID…).

Le système hôte retenu pour transformer les commandes de vol de la cellule de l'Alouette III en une manette de jeu traditionnelle est un Arduino à processeur Atmel AVR et la bibliothèque logicielle open-source LUFA (*Lightweight USB Framework for AVRs*) permettant la mise en œuvre du protocole USB-HID sur les microcontrôleurs Atmel.

Les entrées analogiques de l'Arduino sont numérisées par un CAN 10 bits.

Q14 - L'instrumentation de l'axe 0X avec une alimentation du capteur entre 0 et +5V donne une étendue de mesure EM de +0,15 V à +4,8 V entre les positions extrêmes de la commande de vol. **Quel est le pourcentage** de la pleine échelle couvert par EM ?

Q15 - Dans les mêmes conditions, **quel est l'intervalle** de valeurs obtenu en sortie du CAN ?

En intervention dans des conditions météorologiques normales, le pilote d'un hélicoptère n'agit avec précision sur le manche cyclique qu'avec de très faibles mouvements (il est coutume de dire qu'il déplace la poignée du manche dans un cercle du diamètre d'une pièce de 2 euros…).

Q16 - Toujours pour l'axe longitudinal, **exprimer** le plus petit déplacement de la poignée du manche détectable théoriquement en sortie du CAN.

Tournez la page S.V.P.

2.4. Traitement logiciel

Le programme embarqué, nommé FlightControl, prend en charge une procédure permettant de ramener l'étendue de mesure EM dans un intervalle déterminé.
Cet étalonnage est automatiquement sauvegardé en EEPROM, il ne doit théoriquement être refait qu'après une intervention physique sur les capteurs.

Pour l'Arduino Uno utilisé, la taille de l'EEPROM est de 1 kibioctet.

Q17 - **Que signifie le terme** EEPROM ? Quelles sont les particularités de ce type de circuit mémoire ?

Q18 - **Quelle est la capacité** en octets de l'EEPROM utilisée ?

La bibliothèque Arduino propose les 3 fonctions suivantes publiées dans EEPROM.h :

`int EEPROM.length() ;`	retourne la taille de l'EEPROM en octets
`void EEPROM.write(int addr, byte val ) ;`	écriture de la valeur val (mot de 8 bits non signé) à l'adresse `addr`, sans effet si l'adresse est invalide (*)
`byte EEPROM.read(int addr ) ;`	lecture de l'octet à l'adresse `addr`, retourne `0xff` si l'adresse est invalide (*)

(*) adresse en dehors de l'intervalle `0..length()-1`

Pour chacun des 4 axes, les valeurs extrêmes (minimale et maximale) lues en sortie des CAN sont mémorisées. La lecture d'une sortie de CAN est réalisée par la fonction :

```
uint16_t analogRead( int pin ) ; // lecture d'une sortie CAN avec pin = 0..3
```

Les types de données non standards sont définis de la manière suivante :

```
typedef unsigned char byte ;       // entier non signé 8 bits
typedef unsigned short uint16_t ;  // entier non signé 16 bits
```

Le code de FlightControl contient la définition d'une classe C++ proposant 4 méthodes *inline* afin d'écrire et de lire les valeurs minimales et maximales pour chacun des axes :

```
class FcEeprom
{
  public :
    void writeMin(int axis, uint16_t value ) { … }
    void writeMax(int axis, uint16_t value ) { … }
    uint16_t readMin(int axis ) { … }
    uint16_t readMax(int axis ) { … }
} ;
```

Le tableau suivant présente le schéma d'organisation de la mémoire EEPROM pour les 4 paires de valeurs 16 bits :

adresse	*n° d'axe*	*valeur extrême*	*octet*
0	0 (tangage)	min	poids faible
1			poids fort
2		max	poids faible
3			poids fort
4	1 (roulis)	min	poids faible
5			poids fort
6		max	poids faible
7			poids fort
8	2 (collectif)	min	poids faible
9			poids fort
10		max	poids faible
11			poids fort
12	3 (lacet)	min	poids faible
13			poids fort
14		max	poids faible
15			poids fort

Q19 - **Proposer** une implémentation pour la méthode `writeMin()` chargée de l'écriture en EEPROM de la valeur minimale `value` de l'axe n° `axis`.

Q20 - **Proposer** une implémentation pour la méthode `readMax()` qui retourne la valeur maximale inscrite en EEPROM pour l'axe n° `axis`. Cette méthode doit retourner 0 si l'axe demandé est invalide.

La classe FcInput est utilisée pour modéliser une entrée analogique :

```
class FcInput
{
  public:
    FcInput(int pin, int min = -32768, int max = 32767 ) ;

    void resetRange() {                                  // min/max value resetting
        m_canMin = (uint16_t)CANMAX ;                    // max. CAN value (0x03FF)
        m_canMax = (uint16_t)CANMIN ;                    // min. CAN value (0x0000)
    }

    void setRange(uint16_t min, uint16_t max ) {         // min/max value setting
        m_canMin = min ;
        m_canMax = max ;
    }

    uint16_t calibrate() {                               // min/max value updating
        uint16_t v = analogRead( m_pin ) ;
        if ( v < m_canMin )   m_canMin = v ;
        if ( v > m_canMax )   m_canMax = v ;
        return v ;
    }

    uint16_t canMin() const ;       // min CAN output registered value

    uint16_t canMax() const ;       // max CAN output registered value

    int currentValue() ;            // 16 bits signed axis current value
```


Tournez la page S.V.P.

```
  private:
    int           m_pin ;           // Arduino IO pin number
    int           m_min ;           // min report value (default -32768)
    int           m_max ;           // max report value (default +32767)
    uint16_t      m_canMin ;        // min CAN output value
    uint16_t      m_canMax ;        // max CAN output value
} ;
```

Q21 - **Justifier** le code de la méthode *inline* `resetRange()`.

Le programme principal flightcontrol.ino implémente les traditionnelles fonctions `setup()` et `loop()`. La partie globale du code contient les définitions suivantes :

```
#define NUMAXIS 4

FcInput* axis[NUMAXIS] ;
int axisPin[NUMAXIS] = { 0, 3, 1, 2 } ;
FcEeprom eeprom ;
```

Q22 - **Développer** l'implémentation externe du constructeur de la classe FcInput.

Le diagramme de séquence ci-contre illustre l'initialisation d'un objet FcInput.

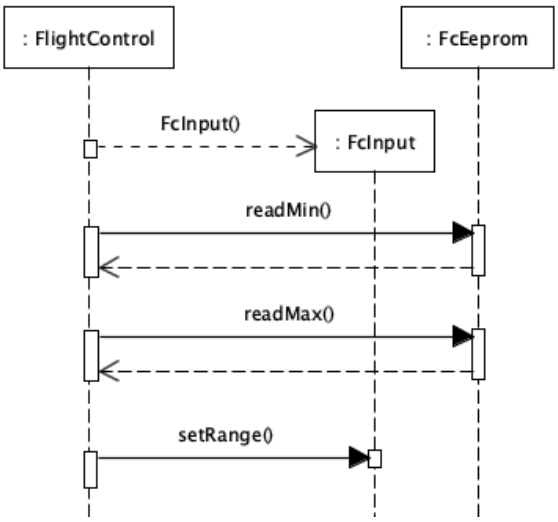


Figure 6 : Initialisation d'un objet FcInput

Q23 - **Proposer** le code nécessaire dans la fonction `setup()` pour la création des 4 instances FcInput et l'initialisation de leur étendue de mesure enregistrée en EEPROM.

L'Arduino est équipé d'un interrupteur INT et d'un témoin lumineux rouge LED1 pour permettre l'autocalibration des 4 axes de commande de vol.

Procédure à suivre :

- Basculer INT en mode Calibrage (position haute) ; le témoin rouge LED1 doit s'allumer ;
- Attention : les valeurs courantes de configuration sont annulées, le système n'est plus opérationnel ;
- Manœuvrer les différentes commandes de vol avec leur débattement maximum (jusqu'aux butées mécaniques), le système scrute en continu les positions extrêmes atteintes ;
- Basculer INT en mode Exploitation ; le témoin LED1 doit s'éteindre. La configuration est enregistrée en mémoire non volatile ; le système est opérationnel.

Q24 - Sur le document réponse DR 24, **compléter** les légendes du diagramme de séquence relatif à la procédure d'autocalibration (simplifié ici pour un unique axe).

Pour une manette de jeu, les grandeurs analogiques sont en général codées sous forme d'une valeur signée sur 16 bits (dans l'intervalle localement maintenu par les membres `m_min` et `m_max` de la classe `FcInput`).
En mode exploitation, les valeurs retournées sur l'interface USB-HID par la méthode `FcInput::currentValue()` sont obtenues par `calibrate()` dans l'intervalle de valeurs `canMin() .. canMax()`, elles doivent ensuite être extrapolées vers la valeur de sortie.

Q25 - **Proposer** une implémentation externe de `FcInput::currentValue()`.

2.5. Conclusion partielle

Q26 - **Conclure** quant aux choix techniques retenus pour la remise en service des commandes de vol de la cellule d'Alouette III.

Partie 3 : Conception du système de projection 3D

Objectifs : Valider la mise en situation du simulateur dans un environnement 3D (écran d'immersion et générateur de scènes).

3.1. Écran de projection

Pour rendre le simulateur le plus réaliste possible, les utilisateurs disposent d'un écran de projection englobant au mieux la cabine de pilotage de l'appareil ; la scène devant être visualisable par le pilote bien entendu, mais aussi par le personnel embarqué afin que celui-ci ressente au mieux les conditions réelles de vol.

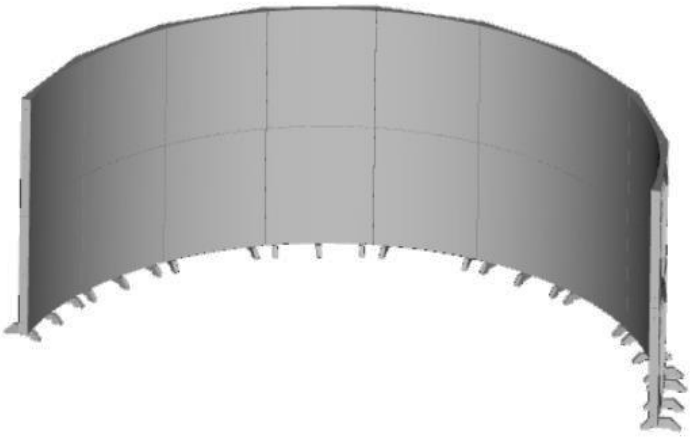


Figure 7 : Écran modulaire semi-cylindrique

Après observation et mesures des parties vitrées et des dimensions de la cellule, il a été envisagé un écran semi-circulaire, de rayon 3 m et d'une hauteur utile égale à 2,40 m.

La structure réalisée en bois est auto-porteuse. Le revêtement de façade est adapté à la projection (finition peinture murale spéciale type « home cinéma »).

Q27 - **Quelle est la surface totale** (en m^2) de l'écran ?

Q28 - **Quel est**, en m, **la longueur** de l'arc de l'ensemble (développé horizontal) ?

L'écran est constitué de deux rangées de 9 panneaux, soit 18 modules identiques. En mode transport, l'ensemble est conditionné sous forme de 6 lots de 3 modules.

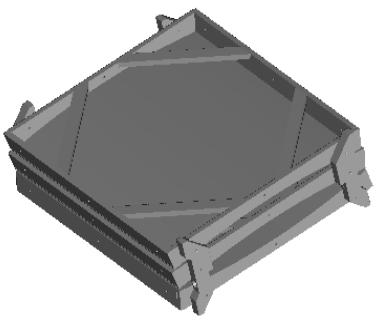


Figure 8 : Lot de transport

Tournez la page S.V.P.

3.2. Projecteurs vidéo

Plusieurs vidéoprojecteurs sont nécessaires pour couvrir la surface totale de l'écran. Celle-ci est totalement couverte par un découpage virtuel en trois formats 4:3 côte à côte, avec un léger recoupement.
Les vidéoprojecteurs utilisés disposent de connexions VGA et HMDI, ils ont pour principales caractéristiques :

- Un rapport de projection 0,6:1 ;
- Une luminosité de 3 300 lumens ;
- Une résolution XGA (cf. DT 3) ;
- Et un contraste de 15 000:1.

Pour libérer l'espace au sol autour de la cellule de l'hélicoptère, les projecteurs sont suspendus tête en bas via une poutre de scène (figure 9).

Figure 9 : Poutre et vidéoprojecteurs

La figure 10 ci-contre montre le positionnement relatif des projecteurs par rapport à l'appareil et à l'écran.

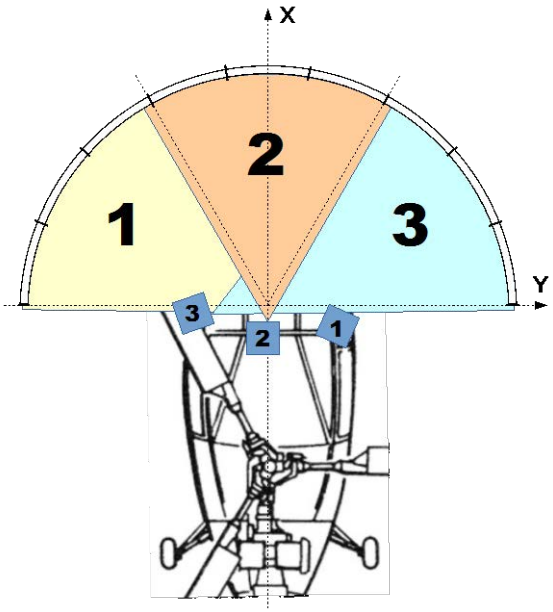


Figure 10 : Projecteurs, vue de dessus

Q29 - **Estimer** la taille d'un pixel projeté sur l'écran panoramique.

On considère qu'un œil normal peut percevoir un détail *d* à une distance *D* lorsque les rayons lumineux issus de ce détail arrivent dans l'œil avec un angle *θ* supérieur ou égal à une minute d'angle (figure 11).

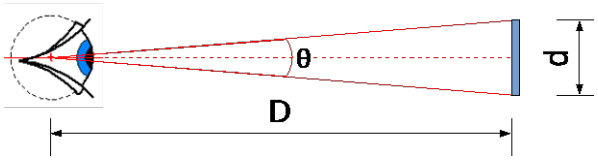


Figure 11 : Détail perceptible par l'œil humain

Q30 - **Calculer** le plus petit détail *d* (mm) observable à l'œil nu à une distance *D* de 3,40 m.

Q31 - **Exprimer** le résultat obtenu précédemment en ppi (*pixel/point par pouce*).

3.3. Générateur de scène 3D

Les scènes 3D sont issues du logiciel X-Plane. Le logiciel propose de faire voler des aéronefs commerciaux ou militaires (avions, hélicoptères, planeurs…) ; il est ici interfacé en USB avec les commandes de vol de la cellule 1790. Des outils sont également fournis pour créer ou personnaliser les aéronefs et les décors.

Figure 12 : Alienware Area-51

La station d'accueil du progiciel simulateur de vol X-Plane 11 (de *Laminar Reseach*) est un Alienware Area-51 équipé notamment d'un processeur Intel® CoreTM i7, de 16 Go de RAM, d'une capacité disque dur de 2 To et de deux unités NVIDIA® GeForce® GTX 980 Ti avec 6 Go de mémoire GDDR5 chacune. Le système d'exploitation est d'origine Windows 7 Pro.
Cette machine héberge également le logiciel Immersive Display Pro (produit par *Fly Elyse-ng*). Le logiciel Immersive Display est utilisé pour le *warping* qui consiste à déformer et raccorder les parties de la scène 3D fournie par X-Plane, scène répartie sur les 3 vidéo-projecteurs.
L'application SceneEditor, qui permet de créer les scénarios d'intervention, est sur ce même poste.

L'application mobile ScenePlayer est utilisée par l'instructeur pour piloter les séances de formations.
La figure ci-dessous résume l'architecture générale du système :

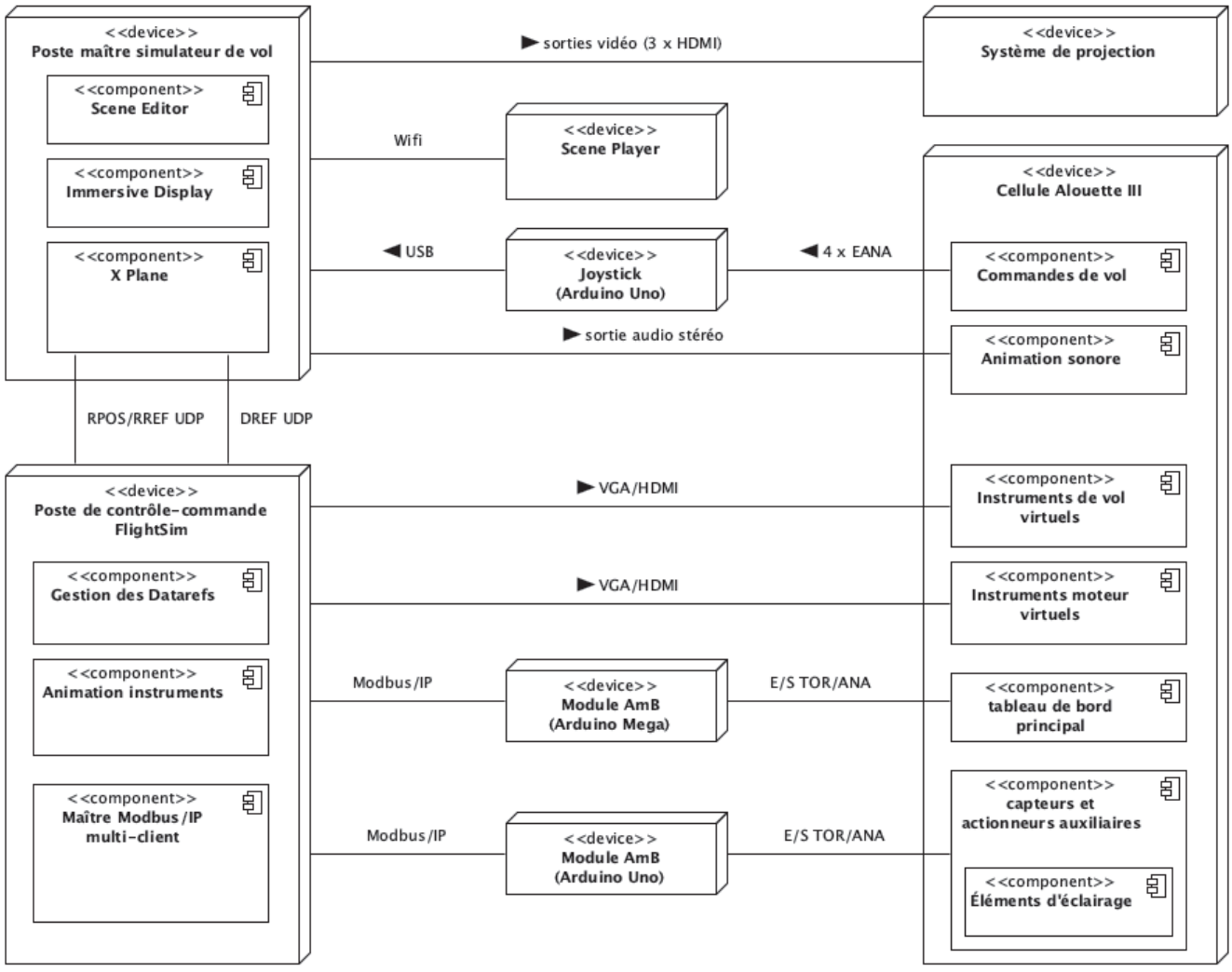


Figure 13 : Diagramme de déploiement général

Q32 - **Conclure** quant à la qualité de la scène observée depuis le cockpit de l'appareil et au choix technique concernant le matériel de projection.

Tournez la page S.V.P.

Partie 4 : Reconstitution du tableau de bord (instruments de vol)

Objectif : Étudier les moyens mis en œuvre pour représenter et animer des instruments de vol en interaction avec un logiciel simulateur de vol.

4.1. Dessin des instruments

La cellule utilisée pour développer le simulateur est dépourvue de tous les instruments de vol d'origine ; ceux-ci sont pour la plupart gyroscopiques ou basés sur la pression statique, ils ne peuvent donc fonctionner qu'en conditions réelles de vol.

Le tableau de bord principal et ses instruments ont donc été reconstitués virtuellement afin de pouvoir être animés en temps réel par les données de vol issues du logiciel X-Plane.

Figure 14 : Tableau de bord réel

La figure 14 présente un exemple de configuration d'origine de la machine SA316B (type de l'Alouette III).
Cette configuration montre notamment l'ensemble « *the standard six* » commun sous diverses formes à tous les aéronefs :

1. Indicateur d'altitude (altimètre) ;
2. Indicateur de vitesse verticale (variomètre) ;
3. Indicateur d'assiette (horizon artificiel) ;
4. Indicateur de vitesse (anémomètre) ;
7. Conservateur de cap ;
9. Indicateur de virage/dérapage (bille).

La pression atmosphérique est la pression exercée par l'air qui entoure la terre. Elle varie selon le moment considéré, la température, la latitude et surtout l'altitude. La pression statique est la pression de l'air au repos ; elle est égale à la pression atmosphérique et est indépendante de la vitesse.

Q33 - **Citer** au moins un des instruments à aiguilles du « *standard six* » qui utilise la pression statique comme grandeur physique d'entrée.

Pour le projet 1790, le tableau de bord a été reconstitué avec une dalle LED (modèle 19M35A de marque LG, VGA, ratio 16:9, résolution 1366 x 768) couverte par un masque modulaire en ABS fabriqué au moyen d'une imprimante 3D.

Cette solution a permis de mettre en place des organes connexes d'entrée/sortie autour des instruments (témoins lumineux, potentiomètres de réglage, boutons-poussoirs…).
Ces organes sont symbolisés figure 15 par les pastilles de couleur.


Figure 15 : Tableau de bord virtuel (instruments représentés non contractuels)

Le graphisme des instruments est généré et animé par une application logicielle dédiée nommée FlightSim, hébergée par la machine hôte (Linux Ubuntu) où est connecté l'écran. Le programme échange des informations sous forme de datagrammes UDP avec le progiciel X-Plane.

Les éléments d'E/S sont gérés par un système embarqué de type Arduino Mega jouant le rôle de serveur *Modbus over TCP/IP*. L'application embarquée est nommée AmB (*Android modipButler*), elle peut gérer jusqu'à 70 E/S dont 16 entrées analogiques.

L'application FlightSim est basée sur la technologie de développement multiplateforme Qt en version *opensource*. Ce framework propose une hiérarchie de classes C++ dédiées au développement d'IHM graphiques.

Avec Qt, le terme générique pour désigner un élément graphique est le widget (*window gadget*) ; les concepts de la programmation orientée objet permettent de spécialiser cette classe d'objets (en l'occurrence QWidget) en une classe spéciale QamFlightInstrument regroupant des propriétés visuelles et dynamiques propres aux instruments de vol.

Chaque instrument est alors modélisé par une classe dédiée, dérivée de la classe QamFlightInstrument (cf. document technique DT 4).

Q34 - Sur le document réponse DR 34, **proposer** une définition de la classe QfiAxis (avec constructeur par défaut et sélecteurs *inline*) en ne retenant pour simplifier que les propriétés 'label' et 'valeur instantanée'.

Q35 - Sur le document réponse DR 35, **développer** le modificateur de la valeur instantanée, en tenant compte des limites admissibles.

Q36 - Sur le document réponse DR 36, **écrire** une définition de la classe QamFlightInstrument à partir des éléments présentés en annexe DT 4, sans les primitives graphiques et avec les mêmes restrictions que précédemment concernant la liste des propriétés retenues pour chaque axe. Toutes les méthodes doivent être externes sauf le constructeur.

Q37 - Sur le document réponse DR 37, **implémenter** le modificateur de valeur instantanée de la classe QamFlightInstrument.

Q38 - Sur le document réponse DR 38, **définir** la classe QamAirSpeedIndicator.

Q39 - **Implémenter** le constructeur de QamAirSpeedIndicator de manière à initialiser l'objet conformément à la figure du document technique DT 4.

Q40 - **Conclure** cette partie par un diagramme UML/SysML (document DR 40) montrant les relations entre les classes QWidget, QfiAxis, QamFlightInstrument et QamAirSpeedIndicator ; détailler la classe QfiAxis (avec les mêmes restrictions qu'en Q34).

4.2. Animation des instruments

Les instruments de vol sont animés en temps réel en fonction des DataRefs fournis par le simulateur de vol X-Plane. Une référence de donnée (*dataref*) peut correspondre à une action (consigne issue des palonniers…), une grandeur instantanée (altitude de l'aéronef…) ou un état (feux de position allumés ou non…).

Les informations souhaitées sont spécifiées et récupérées sous forme de datagrammes UDP (cf. DT 5). Les requêtes ne sont effectuées qu'une fois en début de programme.

La dernière version de X-Plane publie une liste de plus de 4 000 DataRefs classées de manière arborescente ; les grandeurs d'animation des instruments de vol sont par exemple situées dans la catégorie *sim/cockpit2/gauges/indicators/*.

L'animation de l'aiguille de l'ASI (cf. DT 4) est ainsi faite en fonction de la valeur d'une DataRef identifiée par la chaine "*sim/cockpit2/gauges/indicators/airspeed_kts_pilot*".

Qt fournit la classe QByteArray pour gérer les tableaux dynamiques d'octets.
Un objet de cette classe est une suite de données brutes pouvant être alimentée notamment par les méthodes suivantes :

```
QByteArray&     append(const QByteArray& ba ) ;
QByteArray&     append(char ch ) ;
QByteArray&     append(int count, char ch ) ;
QByteArray&     append(const char* str ) ;
QByteArray&     append(const char* str, int len ) ;
QByteArray&     append(const QString& str ) ;
```

La méthode `QamFlightInstrument::xplaneRref()` fabrique le datagramme UDP de requête de la DataRef spécifiée en argument. Son prototype est le suivant :

```
QByteArray xplaneRref(const QString& dataref, int freq, int id ) const ;
```

Q41 - Sachant que le nombre de caractères d'un `QString` peut être obtenu par sa méthode `size()` et que le bourrage (*padding*) est constitué d'octets nuls, **proposer** sur le document réponse DR 41 une implémentation de la méthode `xplaneRref()`.

Q42 - **Que signifie** UDP ? **Quelle est** la couche du modèle OSI concernée ?
Citer un autre protocole de même niveau.

Le document technique DT 6 montre un extrait du fichier de configuration de FlightSim regroupant la liste complète des DataRefs utilisées par les instruments de vol (sur le tableau de bord principal) et les instruments moteurs (sur un second écran) de la cellule Alouette III.
Par exemple, l'altimètre est rafraîchit 10 fois par seconde via des trames de réponse RREF (13 octets) ; alors que l'horloge n'est mise à jour qu'une fois par seconde.

Q43 - En considérant que l'encapsulation Ethernet d'un datagramme UDP nécessite 112 octets, avec un espace intertrame de 96 bits, soit 0,96 µs en Fast Ethernet, **calculer** le débit réseau en bits/s induit par l'animation des 14 instruments.
En déduire le taux de charge théorique (%) lié à cette activité.

Q44 - **Justifier** le choix d'UDP pour répondre au besoin étudié.

4.3. Gestion des éléments actifs

Comme évoqué figure 15, chaque instrument de vol est entouré par un maximum de 4 éléments d'E/S gérés par l'application AmB (serveur *Modbus over TPC/IP*).

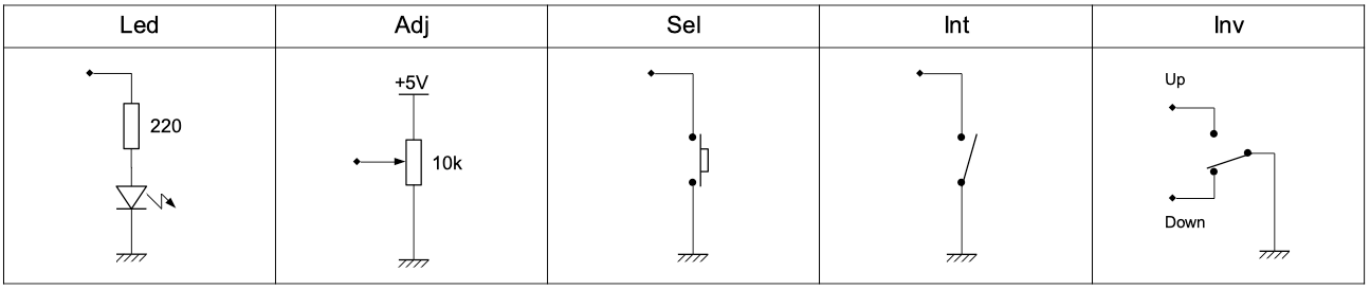
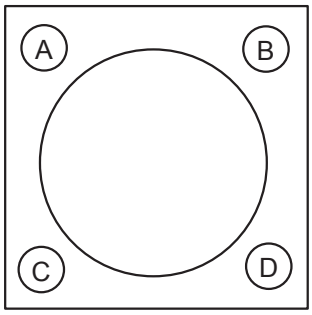


Figure 16 : Types d'éléments d'E/S utilisables

Répartition des éléments d'E/S pour les 8 principaux instruments :



	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
A	Led-R	Led-R	Led-R	Led-R	Led-R	Led-R	Led-R	Sel
B	Led-J	Led-J	Led-J	Led-J	Led-B	Led-B	Led-J	Led-R
C	Adj		Adj	Adj			Adj	Sel
D	Sel	Sel	Sel	Sel	Sel	Sel	Sel	Led-B

note : LEd-R = rouge, Led-J = jaune, Led-B = bleue

Le protocole Modbus supporte les formats 16 bits (*register*) et 1 bit (*coil*) pour l'échange de données. L'encodage retenu pour les E/S du panneau d'instruments est le suivant :

- un registre 16 bits (*input register*) par entrée analogique Adj (l'Arduino dispose de CAN 10 bits). L'esclave Modbus présente toujours les grandeurs en 1/1000e (valeur 0…1000) ;
- un registre 16 bits (*holding register*) pour les entrées-sorties TOR de chaque module. Chaque emplacement A, B, C et D se voit allouer 4 bits : 2 en tant qu'entrée et 2 en tant que sortie. Cette réserve de 4 combinaisons par entrée ou sortie permet notamment d'envisager 4 états « éteint », « allumé fixe », « clignotant lent » et « clignotant rapide » pour les sorties de type Led.

← STOR = Led →								← ETOR = Sel | Int | Inv →							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D		C		B		A		D		C		B		A	

Pour l'altimètre, le sélecteur D est utilisé pour le changement d'unité de pression de l'instrument (hPa ↔ InNg - *inch of mercure*) et le potentiomètre C permet de régler la référence de pression QNH ; en aéronautique, le QNH est la pression barométrique corrigée des erreurs instrumentales (de température et de gravité) et ramenée au niveau moyen de la mer (MSL ou *Mean Sea Level*).

MSL à 15°C : 1013,25 mBar (hPa) = 29,92 InHg

Tournez la page S.V.P.

Les trames suivantes ont été capturées au moyen d'un analyseur de trafic réseau. Elles concernent des transactions Modbus TCP/IP entre l'application cliente FlightSim et le système embarqué AmB. Ces échanges sont relatifs à la gestion des éléments d'E/S de l'altimètre.

```
0000  90 a2 da 10 6f 31 a8 20 66 30 52 a5 08 00 45 00   ....o1. f0R...E.
0010  00 34 00 00 40 00 40 06 00 00 c0 a8 00 05 c0 a8   .4..@.@.........
0020  00 0a f7 11 01 f6 09 d2 2a 45 8e ae 53 0b 50 18   ........*E..S.P.
0030  ff ff 81 86 00 00 00 02 00 00 00 06 ff 04 00 09   ................
0040  00 01                                             ..
```

Figure 17 : Trame 291

```
0000  a8 20 66 30 52 a5 90 a2 da 10 6f 31 08 00 45 00   . f0R.....o1..E.
0010  00 33 e7 e1 40 00 80 06 91 83 c0 a8 00 0a c0 a8   .3..@...........
0020  00 05 01 f6 f7 11 8e ae 53 0b 09 d2 2a 51 50 18   ........S...*QP.
0030  08 00 3a 6e 00 00 00 02 00 00 00 05 ff 04 02 02   ..:n............
0040  dc                                                .
```

Figure 18 : Trame 292

```
0000  90 a2 da 10 6f 31 a8 20 66 30 52 a5 08 00 45 00   ....o1. f0R...E.
0010  00 34 00 00 40 00 40 06 00 00 c0 a8 00 05 c0 a8   .4..@.@.........
0020  00 0a f7 11 01 f6 09 d2 2a 8d 8e ae 53 4d 50 18   ........*...SMP.
0030  ff ff 81 86 00 00 00 08 00 00 00 06 ff 03 00 09   ................
0040  00 01                                             ..
```

Figure 19 : Trame 309

```
0000  a8 20 66 30 52 a5 90 a2 da 10 6f 31 08 00 45 00   . f0R.....o1..E.
0010  00 33 e7 e7 40 00 80 06 91 7d c0 a8 00 0a c0 a8   .3..@....}......
0020  00 05 01 f6 f7 11 8e ae 53 4d 09 d2 2a 99 50 18   ........SM..*.P.
0030  08 00 15 e0 00 00 00 08 00 00 00 05 ff 03 02 02   ................
0040  00                                               .
```

Figure 20 : Trame 310

Consulter les documents techniques DT 7 et DT 8 et analyser les trames précédentes pour répondre aux questions suivantes.

Q45 - **Donner** les adresses IP du poste FlightSim et du serveur AmB en notation décimale pointée.

Q46 - **Quel est** le numéro de port du serveur AmB (en décimal) ?
Est-ce un numéro réservé par l'IANA (*Internet Assigned Numbers Authority*) ? **Justifier.**

Q47 - **Quels sont** les numéros des registres Modbus concernés par les échanges ?

Q48 - **En déduire** la valeur et/ou l'état des éléments d'E/S entourant l'altimètre.

4.4. Synthèse « cellule pilotable »

L'instrumentation des commandes de vol, la reconstitution des instruments et les communications avec le logiciel simulateur de vol ont permis de rendre la vénérable cellule d'Alouette III à nouveau pilotable…

Partie 5 : Architecture réseau du système

Objectifs : Étudier la mise en réseau des différents équipements du simulateur, en local et au sein de l'architecture réseau globale du centre de formation.

5.1. Réseau local

Le réseau local du simulateur d'interventions héliportées est nommé LAN-1790, c'est le premier sous-réseau de 192.168.0.0/27 (notation CIDR).

Q49 - **Qu'est-ce que le CIDR ? Que signifie** /27 dans la notation précédente ? **Pourquoi** le CIDR a-t-il été développé ?

En plus des matériels déjà évoqués, le système 1790 est complété par un deuxième module AmB destiné à la gestion des tableaux de bord auxiliaires (instruments moteur et plafonnier) et par un système de sonorisation relié à la sortie son du poste X-Plane afin de reproduire le bruit de la turbine et autres effets sonores à l'intérieur de la cellule.
En phase d'exploitation, l'instructeur pilote le scénario d'intervention au moyen d'une tablette Android reliée par wifi (application ScenePlayer). Une caméra IP est par ailleurs installée dans la cellule pour permettre à l'instructeur de voir le comportement des occupants et l'état du tableau de bord principal, ceci soit sur l'écran du poste FlightSim, soit directement sur sa tablette.

La liste complète des matériels de LAN-1790 est donc la suivante :

- Poste X-Plane (Windows 7) ;
- Poste FlightSim (Linux) ;
- Système embarqué AmB principal (Arduino Mega) ;
- Système embarqué AmB secondaire (Arduino Uno) ;
- Caméra IP embarquée ;
- Borne wifi ;
- Tablette Android dédiée ;
- Commutateur réseau.

Le réseau LAN-1790 est illustré figure suivante.

Q50 - **Quel est** le masque du réseau LAN-1790 ? **Combien** d'hôtes accepte-t-il ? Avec **quelle plage d'adresses** ? Et **quelle est** son adresse de diffusion ?

Q51 - **Proposer,** sur le document réponse DR 51, un plan d'adressage pour les équipements du réseau LAN-1790.

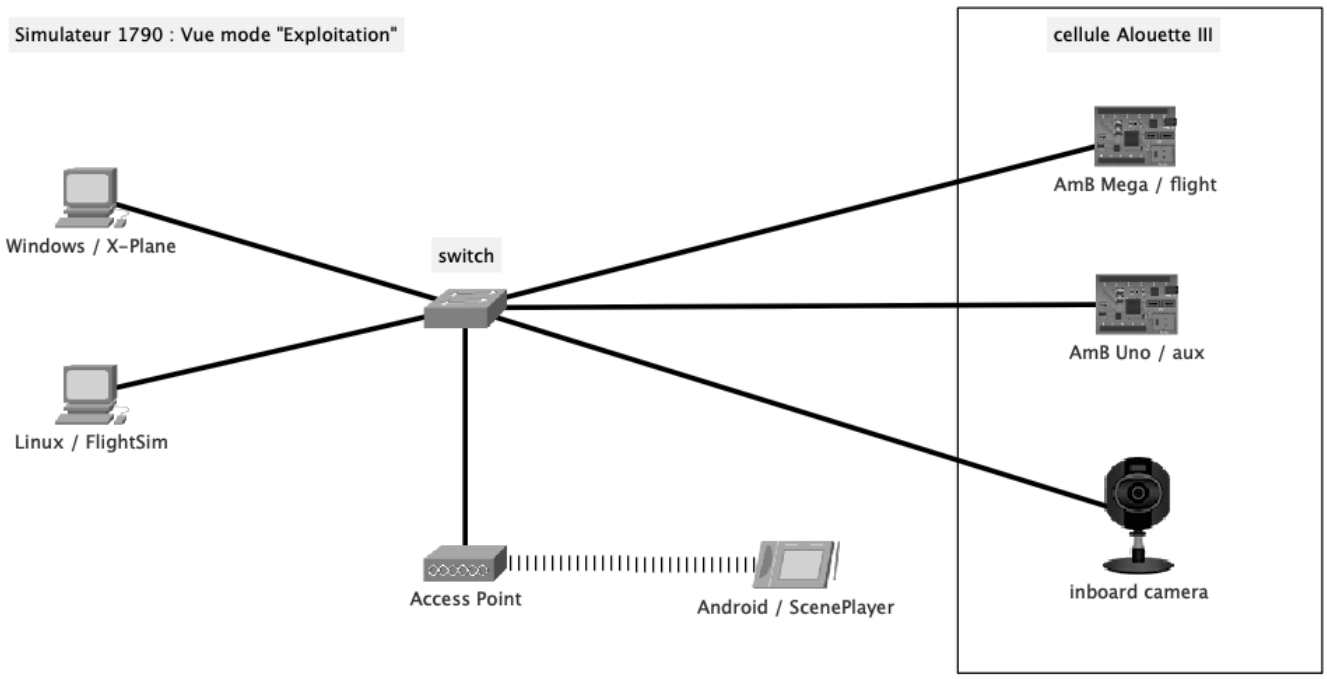


Figure 21 : Réseau local LAN-1790

Le centre de formation qui héberge le simulateur 1790 prévoit à moyen terme la mise en service :

- De 3 autres ateliers pratiques (LAN-TP1, LAN-TP2 et LAN-TP3) nécessitant chacun un maximum de 10 adresses ;
- Et 2 salles de formations spécialisées (LAN-FS1 et LAN-FS2), équipées chacune de 12 postes informatiques, 1 poste formateur, 1 borne wifi, 1 vidéoprojecteur connecté et une imprimante réseau ; chaque salle de formation devant être susceptible d'accueillir ponctuellement jusqu'à 16 postes mobiles (ordinateurs portables, tablettes…).

Les 6 zones de formation sont toutes desservies par le réseau 192.168.0.0/24.

Q52 - **Comment se nomme** la technique qui consiste à répartir des équipements en sous-réseaux de tailles variables ?

Q53 - **Proposer** une répartition en sous-réseaux au plus juste pour chacune des 6 zones du centre de formation (document réponse DR 53) ; **spécifier** aussi les zones d'adresses laissées libres.

5.2. Réseau étendu

L'architecture réseau précédente doit ensuite être intégrée dans un ensemble plus vaste qui regroupe notamment un bâtiment administratif, des serveurs et un accès Internet.

Les équipements des 6 zones de formation (réseau 192.168.0.0/24) seront alors tous connectés à un commutateur unique (constitué de plusieurs éléments 48 ports en cascades).
Les administrateurs prévoient par ailleurs la mise en œuvre de la technologie des VLAN (réseaux locaux virtuel) pour associer les zones de formation avec des machines spécifiques de la partie administrative. Le réseau local LAN-1790 sera par exemple lié à un serveur hébergeant les données d'interventions réelles (comptes-rendus, moyens déployés, banques d'images…) pouvant être utilisées pour la création de scénarios sur le simulateur ; les salles LAN-FS1 et LAN-FS2 seront aussi associées à un serveur spécifique pour tous les supports de cours.

Architecture prévisionnelle :

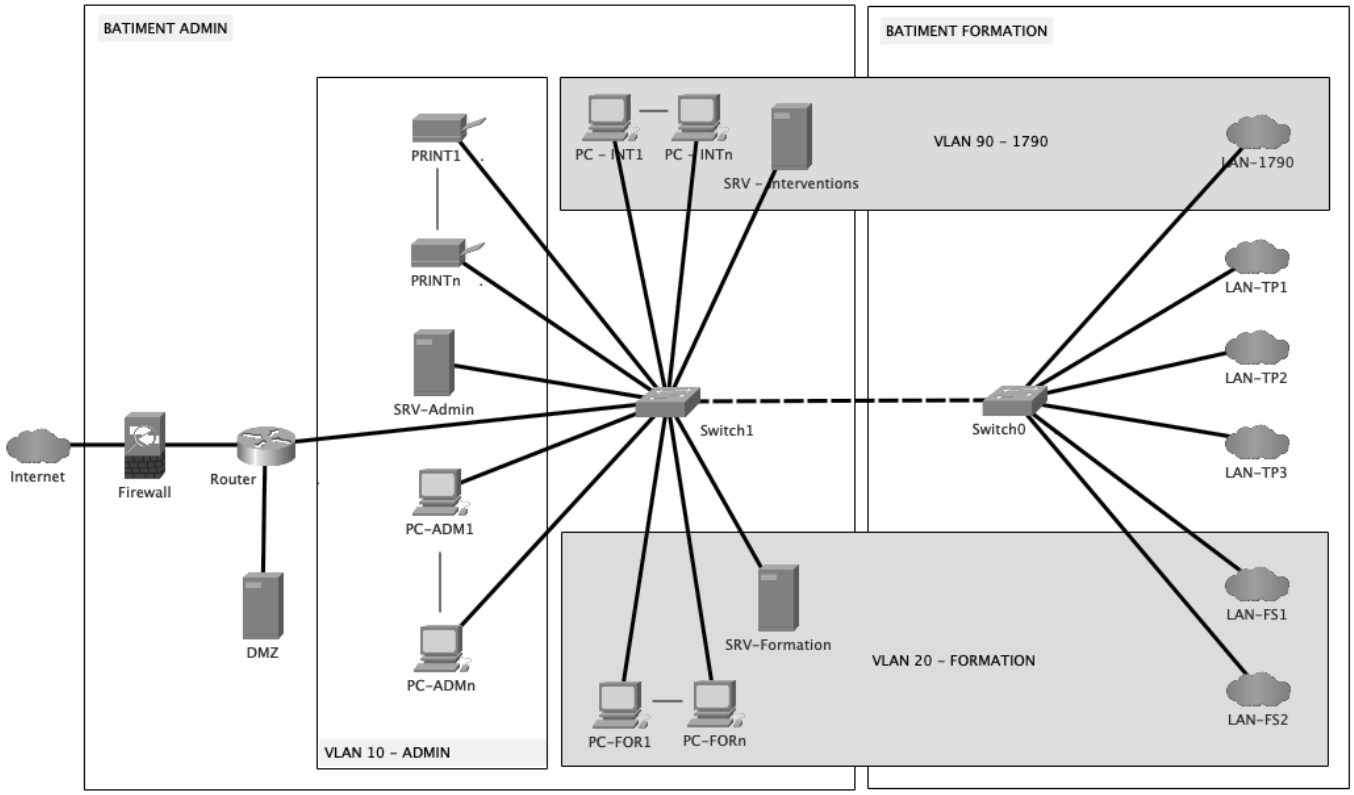


Figure 22 : Réseau global du centre de formation

Q54 - **Quels sont les avantages** des VLAN ?

Un « lien marqué » (*tagged link*) est une interconnexion entre deux commutateurs qui préserve l'appartenance aux VLAN de chaque trame. CISCO et d'autres constructeurs utilisent le terme « lien trunk » (*trunk link*) pour parler d'un lien marqué.
Plusieurs VLAN peuvent partager un même câble d'interconnexion grâce à un système de marquage des trames (*tag*) destiné à transporter leur numéro de VLAN (cf. DT 7, protocole IEEE 802.1Q).

Q55 - **Sur quelle(s) couche(s)** du modèle OSI le marquage des trames est-il effectué ?

Q56 - **Combien** de VLAN distincts le protocole 802.1Q autorise-t-il ?

Q57 - Sur le schéma, le routage inter-VLANs est assuré par l'élément nommé *router* ; **quels sont les éléments** devant être liés par *trunk link* ?

Q58 - Le schéma montre une zone DMZ. **Que signifie** cet acronyme ? **À quoi sert** cette zone ?

Q59 - **Quelle politique de sécurité** est généralement appliquée sur une DMZ (compléter le document réponse DR 59 par des croix) ?

DOCUMENTS TECHNIQUES

DT 1 : Commandes de vol d'un appareil à voilure tournante

Les quatre commandes de vol
Le « manche à balai » permet de modifier l'assiette de l'appareil en jouant sur l'inclinaison du rotor principal ; cet actionneur nommé « commande de pas cyclique » (ou simplement « cyclique ») autorise donc à lui seul le contrôle de tous les mouvements dans le plan horizontal :
1 - axe de tangage (Ry : *pitch*) : manche cyclique, débattement longitudinal Tx.
2 - axe de roulis (Rx : *roll*) : manche cyclique, débattement latéral Ty ;

La commande de pas général, située en général à main gauche, contrôle la pente de déplacement dans le plan vertical, sans modifier l'assiette de l'appareil :
3 - pas général : levier de pas collectif agissant sur le pas des pales du rotor principal.

Le palonnier (constitué de deux pédales en opposition pour pieds gauche et droit) permet de piloter l'orientation du nez de l'appareil :
4 - axe de lacet (Rz : *yaw*) : palonnier agissant sur le pas du rotor anti-couple (RAC) de queue.

Effets dynamiques
Ces quatre axes de commande ont des effets primaires sur la dynamique de l'aéronef, mais peuvent aussi engendrer des effets secondaires :

commande de vol		*effet primaire*	*effet secondaire*
1	cyclique – longitudinal	vers l'avant : → piqué vers l'arrière : → cabrage	en vol en palier, en montée ou en descente avec une vitesse supérieure à la VOM (Vitesse Optimale de Montée) : → trajectoire modifiée dans le sens du manche en vol en descente avec une vitesse inférieure à la VOM (Vitesse Optimale de Montée) : → trajectoire modifiée à l'inverse du manche
2	cyclique – latéral	vers la gauche : → inclinaison à gauche vers la droite : → inclinaison à droite	
3	collectif	vers le haut : → pente de montée vers le bas : → pente de descente	les variations du collectif induisent des couples d'entraînement et de renversement du rotor principal : → compensations au moyen du palonnier
4	palonnier	appui à gauche : → rotation à gauche appui à droite : → rotation à droite	aucun

Coordination « Assiette – Vitesse – Pas général »
Elle a pour but d'adapter constamment la valeur du pas général à l'assiette et à la vitesse pour maintenir une altitude constante (à noter que le pas général est minimal pour la VOM).

- Si la vitesse est inférieure à la VOM :
  - Pour augmenter la vitesse, on pousse le manche vers l'avant et on diminue légèrement le pas général ;
  - Pour diminuer la vitesse, on met le manche en arrière et on diminue momentanément le pas général puis on l'augmente à nouveau.
- Lorsque la vitesse est supérieure à la VOM :
  - Pour augmenter la vitesse, on pousse le manche en avant et on augmente le pas général ;
  - Pour diminuer la vitesse, on met le manche en arrière et on diminue le pas général.

Coordination « Pas général – Palonnier »
Les variations du collectif entraînent des variations des couples d'entraînement et de renversement de l'hélicoptère ; par voie de conséquence, ces variations entraînent une rotation autour de l'axe de lacet :

- Réduction du pas général : on agit sur le palonnier opposé au couple d'entraînement ;
- Augmentation du pas général : on agit sur le palonnier opposé au couple de renversement.

Pour un hélicoptère dont le rotor principal tourne vers la droite (sens horaire vu de dessus, cas de l'Alouette III) :

- Action sur le palonnier gauche → diminution du pas général ;
- Action sur le palonnier droit → augmentation du pas général.

Ces effets sont inversés si le rotor principal tourne vers la gauche

DT 2 : Documentation VISHAY P11VYS10K (*extrait*)

P11, PA11

Vishay Sfernice

Modular Potentiometers with Cermet (P11) or Conductive Plastic Elements (PA11)

FEATURES

- CECC 41300
- GAM T1
- P11 version for industrial and military applications
- PA11 version for professional audio applications
- Trimmer version T11/TA11 (see document No. 51021)
- Miniature module size: 12.5 mm square - low current compatibility
- Five shaft diameters and 12 terminal styles
- Multiple assemblies - up to seven modules
- Shaft and panel sealed version
- Up to twenty-one indent positions
- Switch modules
- Concentric shafts
- Motorized version
- Custom designs

RoHS COMPLIANT

VERSATILE	MODULAR	COMPACT	ROBUST

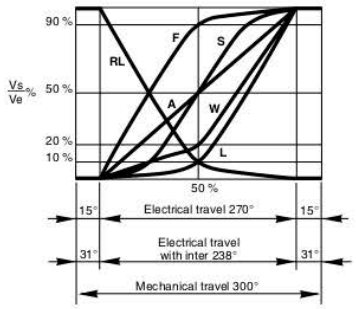
ELECTRICAL SPECIFICATIONS			
		PA11	P11
Resistive Element		Conductive plastic	Cermet
Electrical Travel		270° ± 10°	270° ± 10°
Resistance Range*	Linear Law	1 kΩ to 1 MΩ	10 Ω to 10 MΩ
	Non Linear Law	470 Ω to 500 kΩ	100 Ω to 2.2 MΩ
Tolerance	Standard	± 20 %	± 20 %
	On request	-	± 5 % or ± 10 %
Power Rating	Linear Law	0.5 W at + 70 °C	1 W at + 70 °C
	Non linear Laws	0.25 W at + 70 °C	0.5 W at + 70 °C
	Multiple Assemblies	0.25 W at + 70 °C per module	0.5 W at + 70 °C per module
Temperature Coefficient (Typical)		± 500 ppm/°C	± 100 ppm/°C (R ≥ 100 Ω)
Limiting Element Voltage		350 V	350 V
Contact Resistance Variation	Linear Law	1 %	2 % or 3 Ω
End Resistance (Typical)		2 Ω	2 Ω
Independent Linearity (Typical)	Linear Law	± 5 %	± 5 %
Insulation Resistance		10^6 MΩ min.	10^6 MΩ min.
Dielectric Strength		1500 V_{RMS} min.	1500 V_{RMS} min.
Attenuation		90 dB max. and 0.05 dB min.	-
Mechanical Rotational Life		50 000 cycles	50 000 cycles

* Consult Vishay Sfernice for other ohmic values

MECHANICAL SPECIFICATIONS PA11 AND P11

Mechanical Travel: 300° ± 5°
Operating Torque, Single and Dual Assemblies:
3 mm, 4 mm (1/8") dia. Shafts 0.5 to 1.3 Ncm max. (0.7 to 1.8 oz-inch max.)
6 mm (1/4") dia. Shafts 0.7 to 1.5 Ncm max. (1 to 2.1 oz-inch max.)
Three to Seven Modules (per module): 0.2 to 0.3 Ncm max. (0.3 to 0.45 oz-inch max.)
End Stop Torque:
3 mm, 4 mm (1/8") dia. Shafts 25 Ncm max. (2.1 lb-inch max.)
6 mm (1/4") dia. Shafts 80 Ncm max. (6.8 lb-inch max.)
Tightening Torque:
6 mm, 7 mm (1/4") dia. bushings 150 Ncm max. (13 lb-inch max.)
10 mm (3/8") dia. bushings 250 Ncm max. (21 lb-inch max.)
Weight 7 g to 9 g per module (0.25 to 0.32 oz)

VARIATION LAWS



DT 3 : Tableau des résolutions informatiques standards

Sigle	*Nom*	*Définition (pixels)*	*Format*
CGA	Color Graphics Adapter	320 x 200	1.5
EGA	Enhanced Graphics Adapter	640 x 350	1.8
VGA	Video Graphic Array	640 x 480	4/3=1.33
S-VGA	Super-VGA	800 x 600	4/3=1.33
XGA	eXtra Graphic Array	1024 x 768	4/3=1.33
WXGA	Wide eXtra Graphic Array	1366 x 768	1.77 (16/9)
XGA+	eXtra Graphic Array+	1152 x 864	1.5
WXGA+	Wide eXtra Graphic Array+	1440 x 900	1.6 (16/10)
S-XGA	Super-XGA	1280 x 1024	1.25
HD	HD 720	1280 x 720	16/9
WS-XGA	Wide Super-XGA	1600 x 1024	1.56 (25/16)
S-XGA+	Super-XGA+	1400 x 1050	1.25
WS-XGA+	Wide Super-XGA+	1680 x 1050	1.6 (16/10)
U-XGA	Ultra-XGA	1600 x 1200	1.33
FHD 1080	Full-HD 1080	1920 x 1080	16/9
WU-XGA	Wide U-XGA	1920 x 1200	1.6 (16/10)
QHD	Quad-HD	2560 x 1440	16/9
Q-XGA	Quad-XGA	2048 x 1536	1.33
WQ-XGA	Wide Quad-XGA	2560 x 1600	1.6 (16/10)
4K UHD	4k - Ultra-HD	3840 x 2160	16/9

DT 4 : Classe QamFlightInstrument (Qt/C++)

La classe QamFlightInstrument, dérivée de QWidget, sert de modèle de développement pour des instruments de vol aéronautiques. Son constructeur unique est : `QamFlightInstrument(QWidget* parent = 0 ) ;`

La spécialisation de QamFlightInstrument peut se résumer à surcharger deux méthodes protégées :

- `drawBackground()` qui assure le dessin des parties fixes de l'instrument ;
- `drawForeground()` qui dessine les parties mobiles.

La géométrie de l'instrument (position et dimensions) est gérée par la classe de base QWidget.

La classe QamFlightInstrument prend en charge la gestion d'un maximum de 8 grandeurs réelles ; ces grandeurs jouent le rôle de variables d'animation de l'instrument. Le fichier *header* de la classe contient la ligne suivante :

```
#define     QFI_NUMAXIS 8
```

Chaque grandeur est un objet de classe QfiAxis, celle-ci maintient notamment les propriétés suivantes :

- `label` : un label d'identification facultatif ;
- `unit` : une unité (pourcentage par défaut) ;
- `minimum` : une valeur minimale (0.0 par défaut) ;
- `maximum` : une valeur maximale (100.0 par défaut) ;
- `lowThreshold` : un seuil bas (égal à la valeur minimale par défaut) ;
- `highThreshold` : un seuil haut (égal à la valeur maximale par défaut) ;
- `value` : et sa valeur instantanée.

Par convention, les identifiants respectent la notation *lowerCamelCase*. Une propriété « myProp » est modélisée par un attribut privé `m_myProp`, un sélecteur (accesseur) `myProp()` et un modificateur (mutateur) `setMyProp()`.

Exemple pour la valeur maximale de QfiAxis :

```
float m_maximum ;
float maximum() const { return m_maximum ; }
void setMaximum(float max = 100.0 ) ;
```

Les types de données utilisés sont tous des types standards du C++, les grandeurs numériques sont codées en simple précision. Seules les chaines de caractères sont représentées par le type `QString` offert par la bibliothèque Qt ; le constructeur par défaut de cette classe crée une chaine vide.

Les accesseurs QamFlightInstrument des propriétés précédentes attendent tous un argument supplémentaire nommé *axis* qui représente la grandeur concernée (valeur 0…7). La première grandeur (d'indice 0) est utilisée lorsque cet argument est invalide ou absent.

Primitives graphiques :

QamFlightInstrument propose des ressources permettant de dessiner rapidement certains éléments graphiques de base :

- `qfiBackground()` : un fond de base noir avec anneau effet relief ;
- `qfiAxis()` : un axe d'aiguilles repositionnable (par défaut au centre du widget) ;
- `qfiArc()` : un arc coloré paramétrable ;
- `qfiMarker()` : une graduation de type « *rounded rectangle* » paramétrable ;
- `qfiNeedle()` : une aiguille classique ;
- `qfiText()` : une chaine de caractères positionnée par son centre.

Exemple de spécialisation :

« Air Speed Indicator (ASI) »

classe QamAirSpeedIndicator

grandeur d'animation : axe 0, en noeuds (Knots)

(1 nœud = 1 mille nautique par heure = 1,852 km/h)

Tournez la page S.V.P.

DT 5 : Getting/Sending Data from/to X-Plane (*extraits*)

You can send various different UDP messages TO X-Plane as well, and here are the rules for that:

- X-Plane always receives on port 49000.
- Any strings that you send should be null-terminated!
- Any time you send or receive a structure, the struct alignment must be 4 bytes, or 8 bytes if the struct has any doubles in it!

Below, you will see some variable types that are defined internally to X-Plane, and here they are:

- XCHR (character, in local byte-order for the machine you are on)
- XINT (4-byte int, in local byte-order for the machine you are on)
- XFLT (4-byte ints and floats, in local byte-order for the machine you are on)
- XDOB (double-precision float, in local byte-order for the machine you are on)
- strDIM is 500
- vehDIM is 20

So, now you see some notes about the port to send to, the need to null-term your strings, the struct alignment, and variable names and dimensions that we use. Now you need to know what the format is to send messages to X-Plane by UDP!

All of the UDP messages have the same format, which is:

- 5-character MESSAGE PROLOGUE (to indicate the type of message). The first 4 chars are the message type, the 5th char is a byte of value zero, to null-term the label
- and then a DATA INPUT STRUCTURE (containing the message data that you want to send or receive)

Send me all the Datarefs I want: RREF

Datagram size = 413 byte

0	1	2	3	4	5	6	7	8	9	10	11	12	13 ... 412	
'R'	'R'	'E'	'F'	0	freq (int)				id (int)				DataRef (ASCII string)	*padding* (0)

Where freq is actually the number of times per second you want X-Plane to end this data!
Where id is the integer code you want X-Plane to send back with the dataref value so you can tell which dataref X-Plane is giving you! (since you are likely to ask for MANY different datarefs!)
Where DataRef is the dataref string that you want X-Plane to send to you!

X-Plane will send the message right back to the IP address and port number you sent the RREF command from.

Datagram size = 13 byte

0	1	2	3	4	5	6	7	8	9	10	11	12
'R'	'R'	'E'	'F'	0	id (int)				value (float)			

Where id is the integer code you sent in for this dataref in the struct above.
Where value is the dataref value, in machine-native floating-point value, even for ints!

So, of course, you can send in all the RREF messages you want, to get all the dataref values back that you want.

Send in a freq of 0 to stop having X-Pane send the dataref values.

Set a DataRef to a value: DREF

With this power, you can send in any floating-point value to any data-ref in the entire sim!

Datagram size = 509 byte

0	1	2	3	4	5	6	7	8	9 ... 508	
'D'	'R'	'E'	'F'	0	value (float)				DataRef (ASCII string)	*padding* (0)

Where value is the single-precision float value for the DataRef nul terminated dataref.

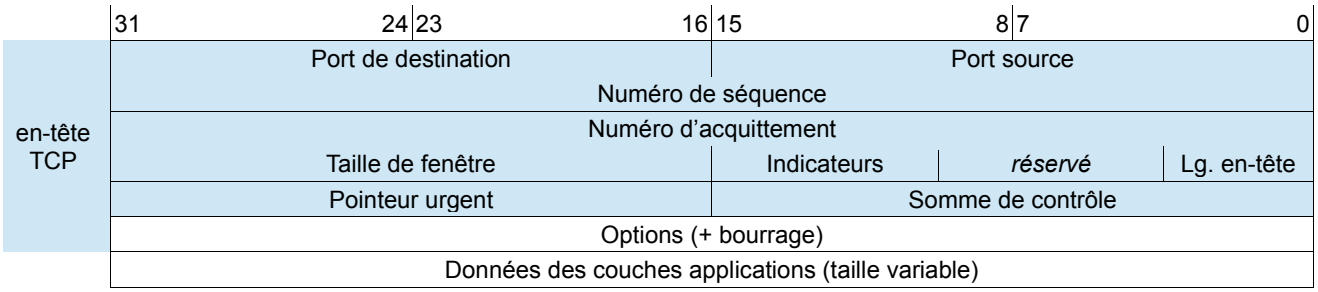
DT 6 : Fichier CSV de configuration de FlightSim (*extrait*)

```
# --------------------------------------------------------------------------
# 0         1              2        3             4     5            6
# RREF_QFI; instrument_id; axis_id; dataref_path; freq; value_scale; value_offset
# --------------------------------------------------------------------------
# Les 2 champs  'value_scale' (default 1)  et 'value_offset' (default 0)  sont
# optionnels.  Ils permettent de mettre en place  le cas échéant  une relation
# linéaire entre la source et la destination :  dest = source * scale + offset
# --------------------------------------------------------------------------

# 1: Altimeter
RREF_QFI; 1; 0; sim/cockpit2/gauges/indicators/altitude_ft_pilot;       10

# 2: VerticalSpeedIndicator
RREF_QFI; 2; 0; sim/cockpit2/gauges/indicators/vvi_fpm_pilot;           10

# 3: ArtificialHorizon
RREF_QFI; 3; 0; sim/cockpit2/gauges/indicators/roll_AHARS_deg_pilot;    20; -1
RREF_QFI; 3; 1; sim/cockpit2/gauges/indicators/pitch_AHARS_deg_pilot;   20

# 4: AirSpeedIndicator
RREF_QFI; 4; 0; sim/cockpit2/gauges/indicators/airspeed_kts_pilot;      15

# 5: OmniBearingIndicator
RREF_QFI; 5; 0; sim/cockpit2/radios/indicators/nav1_hdef_dots_pilot;     5; 4
RREF_QFI; 5; 1; sim/cockpit2/radios/actuators/nav1_obs_deg_mag_pilot;    5
RREF_QFI; 5; 2; sim/cockpit2/radios/indicators/nav1_flag_from_to_pilot;  5

# 6: Chronograph
RREF_QFI; 6; 0; sim/cockpit2/clock_timer/local_time_hours;               1
RREF_QFI; 6; 1; sim/cockpit2/clock_timer/local_time_minutes;             1

# 7: DirectionalGyro
RREF_QFI; 7; 0; sim/cockpit2/gauges/indicators/heading_AHARS_deg_mag;   15

# 8: CollectivePitchIndicator
RREF_QFI; 8; 0; sim/flightmodel/engine/POINT_pitch_deg[0];              20; 0.1
RREF_QFI; 8; 1; sim/cockpit2/temperature/outside_air_temp_degc;         20
RREF_QFI; 8; 2; sim/cockpit2/gauges/indicators/altitude_ft_pilot;       20; 0.3058

# 11: TurnCoordinator
RREF_QFI;11; 0; sim/cockpit2/gauges/indicators/slip_deg;                20; -1

# 21: FuelGauge
RREF_QFI;21; 0; sim/flightmodel/weight/m_fuel[0];                        5; 0.85

# 22: Thermometer
RREF_QFI;22; 0; sim/cockpit2/engine/indicators/EGT_deg_C[0];             5

# 23: Voltmeter
RREF_QFI;23; 0; sim/cockpit2/electrical/battery_voltage_volts[0];        5

# 24: OilGauge
RREF_QFI;24; 0; sim/cockpit2/engine/indicators/oil_temperature_deg_C[0]; 5
RREF_QFI;24; 1; sim/cockpit2/engine/indicators/oil_pressure_psi[0];      5; 0.0689

# 25: Tachymeter
RREF_QFI;25; 0; sim/cockpit2/engine/indicators/engine_speed_rpm[0];     10; 95.7
RREF_QFI;25; 1; sim/cockpit2/engine/indicators/prop_speed_rpm[0];       10

# instrument num = 14, freq sum = 232
```


Tournez la page S.V.P.

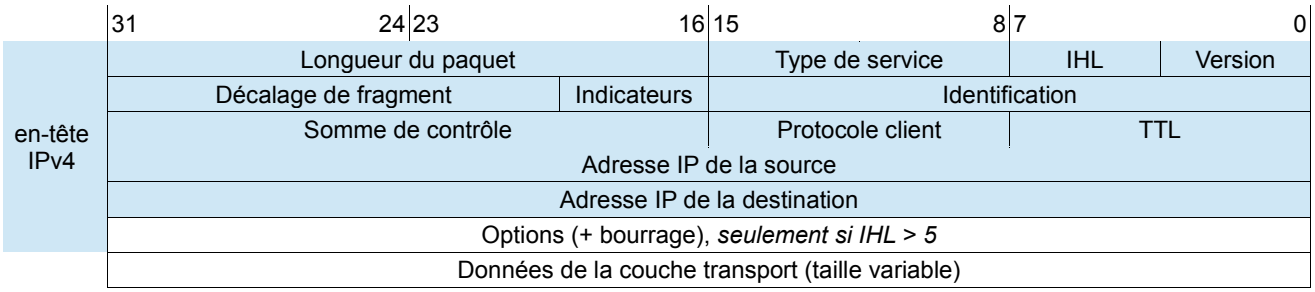
DT 7 : Protocoles TCP / IPv4

Segment TCP



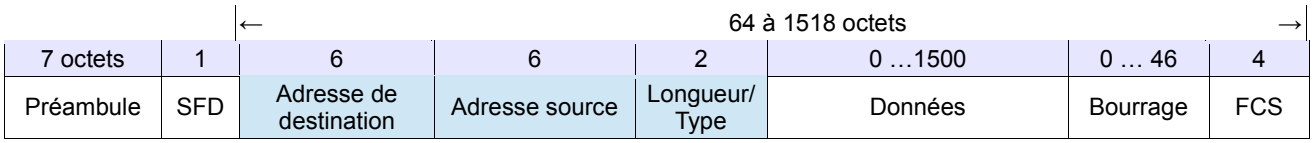
Indicateurs :
- URG = pointeur de données urgente valide
- ACK = champ d'accusé de réception valide
- PSH = fonction de livraison sans attente de remplissage des tampons
- RST = réinitialisation de la connexion
- SYN = synchronisation des numéros d'ordre
- FIN = plus d'envoi de données par l'expéditeur

Paquet IPv4



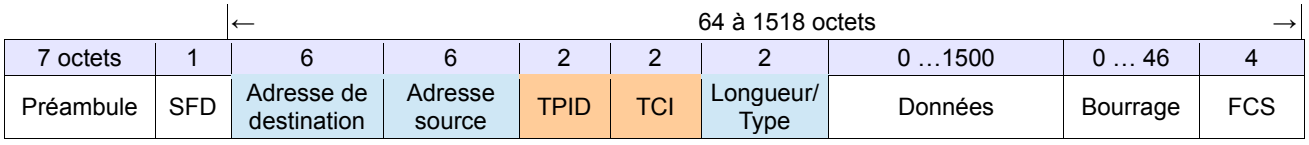
Longueur du paquet (*Total length*) : en octets, minimum = 20 (pas de données)
IHL (*Internet Header Length*) : longueur de l'en-tête, en nombre de mots de 32 bits, minimum = 5
TTL (*Time To Live*) : durée de vie
Protocole client : ICMP = 1, TCP = 6, EGP = 8, IGRP = 9, UDP = 17, EIGRP = 88, OSPF = 89

Ethernet II (standardisé IEEE 802.3)



Préambule : octets `10101010` pour la synchronisation
SFD : *Start Frame Delimiter*, octet `10101011`
Longueur/Type : si ≤ `0x05DC` (1500), 802.3 avec LLC → longueur du champ « Données »
si ≥ `0x0600` (1536), Ethernet II → *EtherType*
`0x0800` = IPv4, `0x86DD` = IPv6, `0x0806` = ARP, `0x8035` = RARP…
FCS : *Frame Check Sequence*, contient un CRC (*Cyclic Redundancy Check*)
Adresses Ethernet : MAC (*Media Access Control*)

Modification IEEE 802.1Q (*VLAN tag*)



TPID : indicateur de protocole = `0x8100`
TCI :
- PCP (3 bits) : *Priority Code Point*, niveau de priorité niveau 2
- CFI (1 bit) : *Canonical Format Identifier*, compatibilité Token Ring, en général à 0
- VID (12 bits) : *VLAN Identifier*, numéro de VLAN, sauf 0 (aucun), 1002…1005 et 4095.

DT 8 : 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.

DOCUMENTS RÉPONSES

(à rendre avec la copie)

Modèle CMEN-DOC v2 ©NEOPTEC

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

Prénom(s) :

Numéro Inscription :

(Le numéro est celui qui figure sur la convocation ou la feuille d'émargement)

Né(e) le : / /

(Remplir cette partie à l'aide de la notice)

Concours / Examen : …….. **Section/Spécialité/Série :** ……

Epreuve : …….. **Matière :** …….. **Session :** ……..

CONSIGNES

- *Remplir soigneusement, sur CHAQUE feuille officielle, la zone d'identification en MAJUSCULES.*
- *Ne pas signer la composition et ne pas y apporter de signe distinctif pouvant indiquer sa provenance.*
- *Numéroter chaque PAGE (cadre en bas à droite de la page) et placer les feuilles dans le bon sens et dans l'ordre.*
- *Rédiger avec un stylo à encre foncée (bleue ou noire) et ne pas utiliser de stylo plume à encre claire.*
- *N'effectuer aucun collage ou découpage de sujets ou de feuille officielle. Ne joindre aucun brouillon.*

EDE NUM 2

DR9 - DR24 - DR34 - DR35

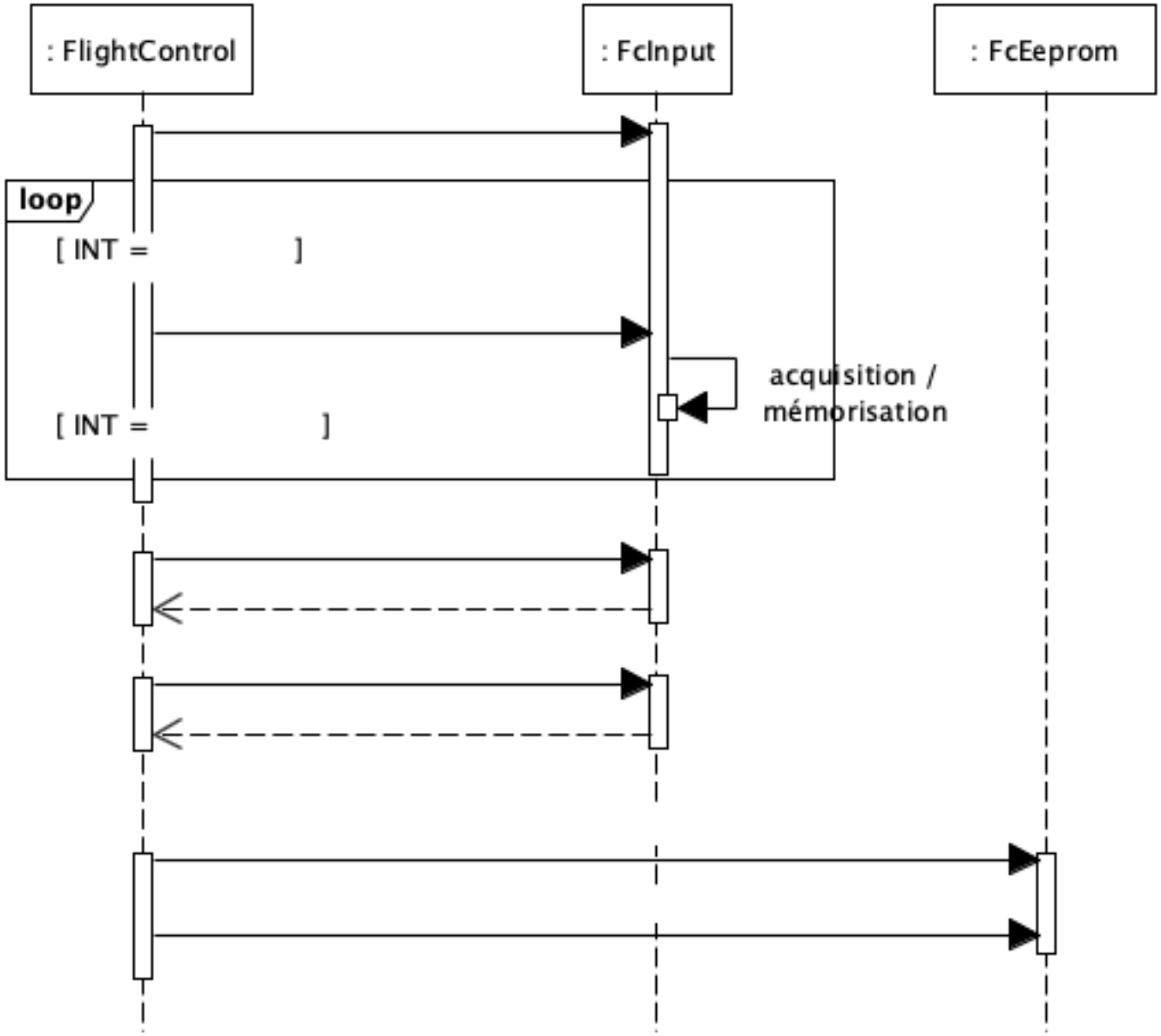
Tous les documents réponses sont à rendre, même non complétés.

NE RIEN ECRIRE DANS CE CADRE

DR 9

degré de liberté	*cyclique*		*collectif*	*palonnier*
	longitudinal	*latéral*		
Tx				
Ty				
Tz				
Rx				
Ry				
Rz				

DR 24



Tournez la page S.V.P.

DR 34

```
#ifndef QFIAXIS_H
#define QFIAXIS_H

class QfiAxis
{
  public :
    // constructeur par défaut

    .....
    .....

    // sélecteurs (accesseurs)

    .....
    .....
    .....
    .....

    // modificateurs (mutateurs)

    .....
    .....
    .....
    .....

  private :
    .....
    .....
    .....
    .....
} ;

#endif
```

DR 35

```
void QfiAxis::setValue( ..................................................... )
{
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
}
```

Modèle CMEN-DOC v2 ©NEOPTEC

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

Prénom(s) :

Numéro Inscription :

(Le numéro est celui qui figure sur la convocation ou la feuille d'émargement)

Né(e) le : / /

(Remplir cette partie à l'aide de la notice)

Concours / Examen : …….. **Section/Spécialité/Série :** ……

Epreuve : …….. **Matière :** …….. **Session :** ……..

CONSIGNES

- *Remplir soigneusement, sur CHAQUE feuille officielle, la zone d'identification en MAJUSCULES.*
- *Ne pas signer la composition et ne pas y apporter de signe distinctif pouvant indiquer sa provenance.*
- *Numéroter chaque PAGE (cadre en bas à droite de la page) et placer les feuilles dans le bon sens et dans l'ordre.*
- *Rédiger avec un stylo à encre foncée (bleue ou noire) et ne pas utiliser de stylo plume à encre claire.*
- *N'effectuer aucun collage ou découpage de sujets ou de feuille officielle. Ne joindre aucun brouillon.*

DR36 - DR37 - DR38 - DR40 - DR41

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

NE RIEN ECRIRE DANS CE CADRE

DR 36

```
#ifndef QAMFLIGHTINSTRUMENT_H
#define QAMFLIGHTINSTRUMENT_H

#include <QWidget>
#include <qfiaxis.h>

class ..........................................................................
{
  public :
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................

  protected :
    virtual void drawBackground() ;
    virtual void drawForeground() ;

  private :
    QfiAxis m_axis[......................] ;
} ;

#endif
```

DR 37

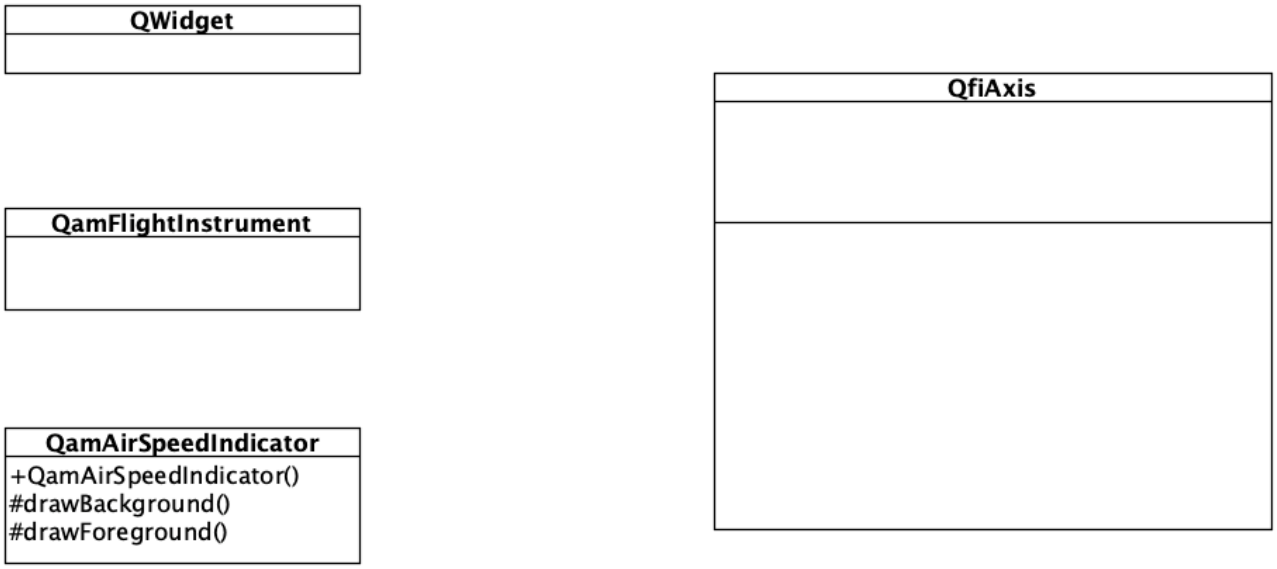
```
void QamFlightInstrument::setValue( ....................................
 ..................................................................... )
{
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
    ..............................................................................
}
```


Tournez la page S.V.P.

DR 38

```
#ifndef QAMAIRSPEEDINDICATOR_H
#define QAMAIRSPEEDINDICATOR_H

#include ..........................................................................

class ..........................................................................
{
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
} ;

#endif
```

DR 40



DR 41

```
QByteArray QamFlightInstrument::xplaneRref( const QString& dataref,
                                            int freq, int id ) const
{
    QByteArray frame ;
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    ..........................................................................
    return frame ;
}
```

Modèle CMEN-DOC v2 ©NEOPTEC

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

Prénom(s) :

Numéro Inscription :

(Le numéro est celui qui figure sur la convocation ou la feuille d'émargement)

Né(e) le : ☐☐ / ☐☐ / ☐☐☐☐

(Remplir cette partie à l'aide de la notice)

Concours / Examen : …….. **Section/Spécialité/Série :** ……

Epreuve : …….. **Matière :** …….. **Session :** ……..

CONSIGNES

- *Remplir soigneusement, sur CHAQUE feuille officielle, la zone d'identification en MAJUSCULES.*
- *Ne pas signer la composition et ne pas y apporter de signe distinctif pouvant indiquer sa provenance.*
- *Numéroter chaque PAGE (cadre en bas à droite de la page) et placer les feuilles dans le bon sens et dans l'ordre.*
- *Rédiger avec un stylo à encre foncée (bleue ou noire) et ne pas utiliser de stylo plume à encre claire.*
- *N'effectuer aucun collage ou découpage de sujets ou de feuille officielle. Ne joindre aucun brouillon.*

EDE NUM 2

DR51 - DR53 - DR59

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

NE RIEN ECRIRE DANS CE CADRE

DR 51

Équipements extérieurs		*Équipements embarqués*	
Poste X-Plane	192.168.	AmB principal	192.168.
Poste FlightSim	192.168.	AmB secondaire	192.168.
Borne wifi	192.168.	Caméra IP	192.168.
Tablette Android	192.168.		

DR 53

nom	*CIDR*	*nb. d'hôtes*	*plage d'adresses des hôtes*	
LAN-1790	192.168.0.0/27		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.
	192.168.0.		192.168.0.	192.168.0.

DR 59

trafic de…	*… vers*	*autorisé*	*interdit*
réseau externe	DMZ		
réseau externe	réseau interne		
réseau interne	DMZ		
réseau interne	réseau externe		
DMZ	réseau interne		
DMZ	réseau externe		