

Étude d'un système, d'un procédé ou d'une organisation

Ingénierie informatique

A. Présentation de l'épreuve

Durée : 4 heures

Coefficient 1

L'épreuve est spécifique à l'option choisie.

L'épreuve a pour but de vérifier que le candidat est capable de conduire une analyse critique de solutions technologiques et de mobiliser ses connaissances scientifiques et technologiques pour élaborer et exploiter les modèles de comportement permettant de quantifier les performances d'un système ou d'un processus lié à la spécialité et définir des solutions technologiques.

B. Sujet

Le sujet est disponible en téléchargement sur le site du ministère à l'adresse :

https://media.devenirenseignant.gouv.fr/file/capet_externe/45/7/s2021_capet_externe_sii_informatique_2_1390457.pdf

SIMULATEUR D'INTERVENTIONS HÉLIPORTÉES



C. Éléments de correction

Partie 1 : Impact socio-économique

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

→ voiture : meilleur cas : $65 / 60 = 1,08 \text{ h} = 1\text{h}05$ – pire cas : $65 / 45 = 1,44 \text{ h} = 1\text{h}26$ ou +...

→ hélico : $56 / 4 = 14 + 10 = 24$ minutes

→ bénéfice : de 41 (bénéfice minimal) à 62 minutes

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.

→ 2000 euros/h → 65 km à parcourir → entre $\approx 2890 \text{ €}$ (à 45 km /h) et 2170 € (à 60km/h)

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...).

→ $30 + 24$ minutes = 54 minutes = 0,9 h

→ $3\ 600 \times 0,9 = 3\ 240 \text{ €}$ pour le vol + forfait décollage = 5 240 €

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

→ taux de charge de l'ordre de 95 % d'après le texte de présentation

→ $24 \times 7 \times 5 \% = 8,4$ heures → 8 heures

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).

→ appareil 4 h/semaine

→ 32 heures de formation => 8 séances pour 3 personnes

→ 12 personnes => 32 séances (soit un étalement sur 32 semaines)

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

Malgré un surcoût de plusieurs milliers d'euros (sans compter les frais de maintenance) par intervention, la réduction importante du temps de prise en charge à l'hôpital donne l'avantage aux moyens aéroportés. Une vie humaine n'a pas de prix...

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

L'utilisation des appareils réels pour la formation n'est que peu envisageable vu leur taux de charge. Le simulateur présente l'avantage d'être disponible en permanence pour l'organisation de sessions de formation.

Partie 2 : Instrumentation des commandes de vol

2.1. Commandes d'une voilure tournante

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

- compenser le couple d'entraînement en rotation induit par le rotor principal sur la cellule
- compensation modulable pour contrôler l'orientation du nez de l'appareil

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

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

2.2. Mise en place des capteurs

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).

- course rotative : 90% de 270° (voir doc. VISHAY P11VYS10K) $\approx 240^\circ$
- diamètre poulie : $D \geq L \cdot 360 / (240 \cdot \pi)$
- $D \geq L \cdot 3 / (2 \cdot \pi)$

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

- d'après le schéma, débattement de $2 \times 15^\circ$ ($2 \times \pi/12$)
- $\sin(\pi/12) = (L/2) / h$
- $L = 2 \cdot h \cdot \sin(\pi/12)$
- avec $h = 158$ mm, $L \approx 82$ mm

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

- $D \geq L \cdot 3 / (2 \cdot \pi) \rightarrow D \geq 39$ mm

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) ?

- lié au battement vertical de la bielle : $E = h \cdot (1 - \cos(\pi/12)) \approx 5,4$ mm
- distance minimale = $5,4 + D/2 = 5,4 + 19,5 = 24,9$ mm

2.3. Acquisition des grandeurs physiques

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 ?

→ pleine échelle = 0..5 V

→ EM = 4,8 – 0,15 = 4,65 V soit 93 % de la pleine échelle

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

→ CAN 10 bits : sortie 0..1023 pour entrée 0..5 V

→ suivant EM : de $0,15 \cdot 1023 / 5 = 30$ à $4,8 \cdot 1023 / 5 = 982$

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

→ quantum (variation minimale de l'entrée pour incrémenter ou décrémenter la sortie du CAN) = $(V_{\max} - V_{\min}) / (N_{\max} - N_{\min}) = 5 / 1023 \approx 4,89 \text{ mV}$

→ rapport course du potentiomètre : $0,00488 / 4,65 \approx 1/1000$

→ la relation entre l'angle du curseur du potentiomètre et l'angle du manche est linéaire.

→ battement d'angle du manche = $1/1000$ de $30^\circ = 0,03^\circ$

→ poignée située à H = 650 mm de l'axe de rotation du manche → $650 \cdot \sin 0,03^\circ \approx 0,34 \text{ mm}$

2.4. Traitement logiciel

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

→ Electrically-Erasable Programmable Read-Only Memory,

→ mémoire morte programmable, non volatile et effaçable électriquement.

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

→ 1 kio = 2^{10} octets = 1024 octets

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

```
void writeMin(int axis, uint16_t value )
{
// if (( axis < 0 )|| ( axis > 3 )) return ;
EEPROM.write( axis * 4, value & 0xFF ); // value % 256
EEPROM.write( axis * 4 + 1, ( value >> 8 ) & 0xFF ); // value / 256
}
```

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.

```
uint16_t readMax(int axis )
{
if (( axis < 0 )|| ( axis > 3 )) return 0 ;
return EEPROM.read( axis * 4 + 3 ) << 8 | EEPROM.read( axis * 4 + 2 );
}
```

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

→ pour `m_canMin`, il faut successivement conserver la plus petite valeur lue par la méthode `calibrate()`, il est donc nécessaire d'initialiser la mémoire avec la valeur extrême opposée. Même démarche inversée pour `m_canMax`.

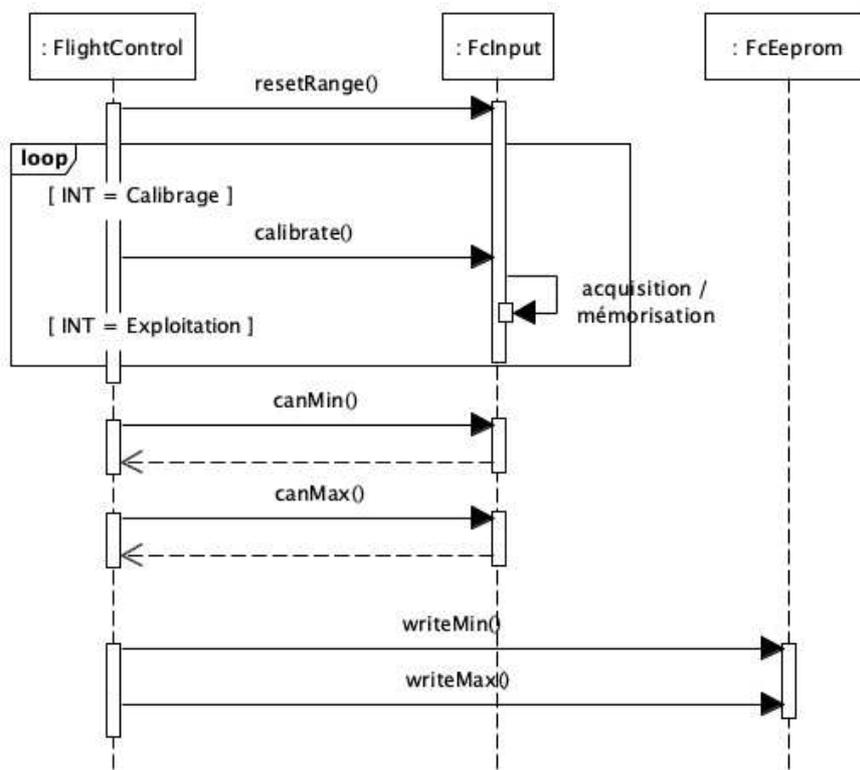
Q22 - Développer l'implémentation externe du constructeur de la classe `FcInput`.

```
FcInput::FcInput(int pin, int min, int max )
    : m_pin( pin)
    , m_min(min)
    , m_max(max)
{
    resetRange();
}
```

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.

```
for ( int i = 0 ; i < NUMAXIS ; ++i ) {
    axis[i] = new FcInput( axisPin[i] ) ;
    axis[i] ->setRange(eeprom.readMin(i), eeprom.readMax(i) ) ;
}
```

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).



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

```
int FcInput::currentValue()
{
    uint16_t fromRange = canMax() - canMin();
    float toRange = m_max - m_min;
    float scale = toRange / fromRange;

    return (int)( ( calibrate() - canMin() ) * scale + m_min );
}
```

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.

→ choix globalement très économiques (partie mécanique de mise en position des capteurs non évaluable).

→ capteurs : solution robuste et pérenne, maintenance aisée

→ système numérique : mise en œuvre simple

Partie 3 : Conception du système de projection 3D

3.1. Écran de projection

Q27 - | Quelle est la surface totale (en m²) de l'écran ?

→ rayon : $R = 3\,000\text{ mm}$

→ amplitude angulaire : $\alpha = 180^\circ$

→ hauteur utile : $H = 2\,400\text{ mm}$

→ surface totale de l'écran : $S = H \cdot \pi \cdot R$ soit $S \approx 22,6\text{ m}^2$

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

→ développé horizontal égal à : $L = \pi \cdot R$ soit $L \approx 9,425\text{ m}$

3.2. Projecteurs vidéo

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

→ 1 vidéoprojecteur couvre 3 x 2 modules d'écran (en négligeant la marge de recouvrement), soit une largeur égale à la corde de 3 panneaux alignés horizontalement ($2 \cdot R \cdot \sin(\pi/6) = 3,00\text{ m}$) et une hauteur de 2,40 m.

$3000 / 1024 = 2,93\text{ mm}$

$2400 / 768 = 3,125\text{ mm}$

→ estimation taille pixel $\approx 3 \times 3\text{ mm}$

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

→ $\theta = 1$ minute d'angle = $1/60^\circ$ de degré

→ $\theta / 2$ degrés = $(\theta / 2) \cdot (2\pi / 360)$ radians

→ $d = 2 \cdot D \cdot \tan(\theta \cdot \pi / 360)$ → $d = 2 \cdot 3400 \cdot \tan(\pi / 21600) \approx 0,99\text{ mm}$

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

→ 1 pouce = 25,4 mm

→ 0,99 mm correspond à 25 ppi (arrondi entier).

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.

→ qualité moyenne : la scène apparaît pixélisée !

→ d'après l'annexe 3, XGA n'est pas la meilleure résolution en format 4:3, le choix de ce type de produit est probablement un compromis qualité/prix.

→ des projecteurs Q-XGA permettraient d'améliorer la qualité de l'image tout est respectant la contrainte de format (mais avec un budget plus élevé...).

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

4.1. Dessin des instruments

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

→ 1 : altimètre, 2 : variomètre, 4 : anémomètre

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'.

```
#ifndef QFIAXIS_H
```

```
#define QFIAXIS_H
```

```
class QfiAxis
```

```
{
```

```
public :
```

```
    // constructeur par défaut
```

```
    QfiAxis() : m_min(0.0), m_max(100.0), m_value(0.0) {}
```

```
    // sélecteurs (accesseurs)
```

```
    QString label() const { return m_label ; }
```

```
    float value() const { return m_value ; }
```

```
    // modificateurs (mutateurs)
```

```
    void setLabel(const QString& label ) ;
```

```
    void setValue(float value ) ;
```

```
private :
```

```
    QString    m_label ;
```

```
    float      m_value ;
```

```
};
```

```
#endif
```

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

```
void QfiAxis::setValue(float value )
{
    if ( value < minimum() )    value = minimum() ;
    if ( value > maximum() )    value = maximum() ;
    m_value = value ;
}
```

Q36 - Sur le document réponse DR 36, écrire une définition de la classe QamFlightInstrument à partir des éléments présentés en annexe, 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.

```
#ifndef QAMFLIGHTINSTRUMENT_H
#define QAMFLIGHTINSTRUMENT_H

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

class QamFlightInstrument : public QWidget
{
public :
    QamFlightInstrument(QWidget* parent = 0 ) : QWidget(parent) {}

    QString label(int axis = 0 ) const ;
    float value(int axis = 0 ) const ;

    void setLabel(const QString& label, int axis = 0 ) ;
    void setValue(float value, int axis = 0 ) ;

protected :
    virtual void drawBackground() ;
    virtual void drawForeground() ;
private :
    QfiAxis    m_axis[ QFI_NUMAXIS ] ;
};

#endif
```

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

```
void QamFlightInstrument::setValue(float value, int axis )
{
    if (( axis < 0 ) || ( axis > QFI_NUMAXIS - 1 )) axis = 0 ;
    m_axis[ axis ].setValue( value ) ;
}
```

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

```
#ifndef QAMAIRSPPEEDINDICATOR_H
#define QAMAIRSPPEEDINDICATOR_H

#include <qamflightinstrument.h>
```

```

class QamAirSpeedIndicator : public QamFlightInstrument
{
public :
    QamAirSpeedIndicator(QWidget* parent = 0 ) ;

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

#endif

```

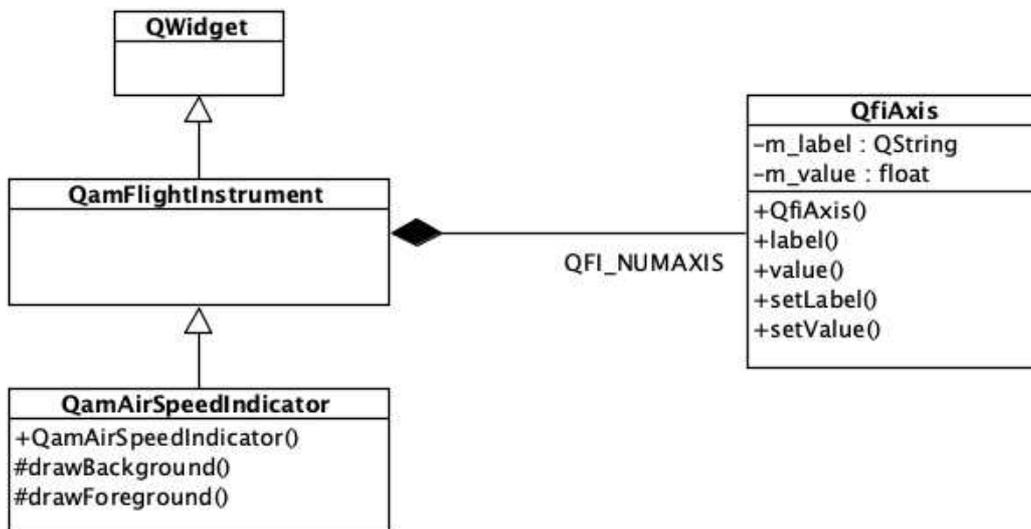
Q39 - Implémenter le constructeur de QamAirSpeedIndicator de manière à initialiser l'objet conformément à la figure du document technique DT 4.

```

QamAirSpeedIndicator:: QamAirSpeedIndicator(QWidget* parent )
: QamFlightInstrument( parent )
{
    setLabel( "ASI" ) ;
    setMinimum( 0.0 ) ;
    setMaximum( 140.0 ) ;
}

```

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

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().

```

QByteArray QamFlightInstrument::xplaneRref(const QString& dataref, int freq, int id ) const
{
    QByteArray frame ;
    frame.append( "RREF", 4 ) ;
    frame.append( char(0) ) ;
}

```

```

frame.append( (char*)&freq, 4 );
frame.append( (char*)&id, 4 );
frame.append( dataref );
frame.append( QByteArray( 400 - dataref.size(), 0 ) );
return frame ;
}

```

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

- User Datagram Protocol
- couche 4 (Transport)
- TCP (Transmission Control Protocol)

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.

- 13 octets par datagramme UDP, soit 13 + 112 = 125 octets (1 000 bits par trame)
- soit en tenant compte des intertrames, 1 096 bits par trame
- nombre total de datagrammes par seconde = somme des champs 'freq' = 232
- soit 254 272 bits / seconde
- à 100 Mb/s (Fast Ethernet), taux de charge = 0,25 %

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

- moins fiable que TCP mais dégradation de trafic supportable ponctuellement pour l'animation des aiguilles des instruments ;
- plus rapide que TCP (pas d'aller-retour de contrôle, entête de trame plus courte) ; donc animations potentiellement plus fluides avec une charge réseau minimale.

4.3. Gestion des éléments actifs

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

- adresse IP du poste FlightSim = (c0 a8 00 05) 192.168.0.5
- adresse IP du serveur AmB = (c0 a8 00 0a) 192.168.0.10

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.

- port du serveur AmB = (01 f6) 502
- numéro de port réservé car < 1024

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

- trame 291 : fonction 4 (*read analog input registers*), adresse 0009, nombre 1
 - registre « input register » numéro 30010
- trame 309 : fonction 3 (*read analog output holding registers*), adresse 0009, nombre 1
 - registre « holding register » numéro 40010

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

- le registre 30010 concerne le réglage Adj, valeur retournée = 02dc soit 732
- le registre 40010 concerne Led-R, Led-J et Sel, valeur retournée = 0200
- Led-R = état « clignotant lent », Led-J = état « éteinte », Sel = 0 (probablement fermé)

Partie 5 : Architecture réseau du système

5.1. Réseau local

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

→ CIDR : *Classless InterDomain Routing*

Le CIDR est basé sur l'idée de masque de sous-réseau. Un masque est placé sur une adresse IP et crée un sous-réseau. Le masque de sous-réseau indique au routeur quelle partie de l'adresse IP détermine le réseau et quelle partie est attribuée aux hôtes.

192.168.0.0/27 est un exemple de notation CIDR où le préfixe /27 est le nombre de bits de la partie réseau en partant de la gauche de la notation décimale pointée sur 32 bits (4 octets). Les N bits restants permettent 2^N adresses ; la première (tous les bits à 0) représente le sous-réseau, la dernière (tous les bits à 1) est l'adresse de diffusion (*broadcast*) du sous-réseau et les $2^N - 2$ restantes sont pour les hôtes.

→ Internet est énorme ; le système d'adresses IP originel (IPv4) est épuisé depuis longtemps. Toutes les adresses possibles ont déjà été attribuées. Il a fallu trouver des solutions il y a quelques décennies pour résoudre le problème. Le CIDR aide à augmenter le nombre d'adresses disponibles, en attendant la généralisation de IPv6...

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 ?

- masque 255.255.255.224
- nombre d'hôtes = $2^5 - 2 = 30$
- plage = 192.168.0.1 ... 192.168.0.30
- broadcast = 192.168.0.31

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

→ tous les équipements en IP statique, dans la plage du sous-réseau .0.1 à .0.30.

Équipements extérieurs		Équipements embarqués	
Poste X-Plane	192.168.0.1	AmB principal	192.168.0.10
Poste FlightSim	192.168.0.5	AmB secondaire	192.168.0.11
Borne wifi	192.168.0.30	Caméra IP	192.168.0.20
Tablette Android	192.168.0.7		

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

→ VLSM : *Variable Length Subnet Masking*

VLSM est la partie du CIDR qui permet de réaliser des sous-réseaux de longueur variable. Les préfixes utilisables pour un réseau de masque 255.255.255.0 (24 bits) vont de /25 (2 sous-réseaux de 126 hôtes) à /30 (64 sous-réseaux de 2 hôtes).

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.

<i>nom</i>	<i>CIDR</i>	<i>nb. d'hôtes</i>	<i>plage d'adresses</i>	
LAN-1790	192.168.0.0/27	30	192.168.0.1	192.168.0.30
LAN-TP1	192.168.0.32/28	14	192.168.0.33	192.168.0.46
LAN-TP2	192.168.0.48/28	14	192.168.0.49	192.168.0.62
LAN-TP3	192.168.0.64/28	14	192.168.0.65	192.168.0.78
<zone libre>	192.168.0.80/28	14	192.168.0.81	192.168.0.94
<zone libre>	192.168.0.96/27	30	192.168.0.97	192.168.0.125
LAN-FS1	192.168.0.128/26	62	192.168.0.129	192.168.0.190
LAN-FS2	192.168.0.192/26	62	192.168.0.193	192.168.0.254

5.2. Réseau étendu

Q54 - Quels sont les avantages des VLAN ?

- Améliorer la gestion du réseau ;
- Optimiser la bande passante ;
- Séparer les flux ;
- Segmentation : réduire la taille d'un domaine de *broadcast* ;
- Sécurité : permet de créer un ensemble logique isolé pour améliorer la sécurité. Le seul moyen pour communiquer entre des machines appartenant à des VLAN différents est alors de passer par un routeur.

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

- protocole Ethernet = couche « accès réseau » du modèle TCP/IP
- équivalence : couches 1 (physique) et 2 (liaison de données) du modèle OSI

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

- champ VID sur 12 bits avec valeurs 0, 1002 à 1005 et 4095 réservées
- $2^{12} - 6 = 4090$

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* ?

- Switch0 et Switch1

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

→ DMZ = *DeMilitarized Zone*, zone démilitarisée

→ sous-réseau séparé du réseau local et isolé de celui-ci et d'Internet par un pare-feu ; utilisé pour restreindre les accès extérieurs à des services locaux choisis (serveur Web, serveur de messagerie...).

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

<i>trafic de...</i>	<i>... vers</i>	<i>autorisé</i>	<i>interdit</i>
réseau externe	DMZ	X	
réseau externe	réseau interne		X
réseau interne	DMZ	X	
réseau interne	réseau externe	(X)	X
DMZ	réseau interne	X	
DMZ	réseau externe	X	