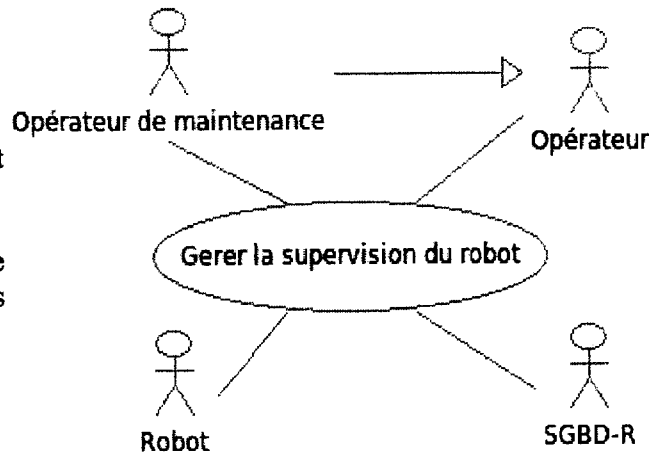


C. Contrôle-Commande et Supervision

C.1. Analyse du système de supervision

Le système de supervision du robot interagit avec :

- le robot,
- un système de gestion de base de données permettant d'enregistrer les points de passage du robot,
- l'opérateur du système,
- et l'opérateur de maintenance.



Ceci est présenté dans le diagramme ci-contre.

Cette analyse des cas d'utilisation est sommaire, elle permet de définir les limites du système. Ceci implique, par exemple, que le robot est considéré comme acteur du système de supervision...

Question C.1.1.

Que signifie la flèche reliant Opérateur de maintenance à Opérateur ?

Question C.1.2.

Le document réponse propose quatre diagrammes de cas d'utilisation d'un niveau de détail supérieur à celui ci-dessus. Deux d'entre eux sont manifestement incohérents avec le diagramme ci-dessus. Indiquer lesquels et justifier la réponse.

C.2. Modes de fonctionnement du robot

Comme évoqué précédemment, le robot possède au moins deux modes de fonctionnement : un mode automatique dans lequel il gère seul son déplacement, et un mode de pilotage manuel se traduisant, vu de l'opérateur, par des envois d'ordres (position à atteindre, arrêt, prélèvement,...) et de commandes de déplacement (par l'intermédiaire d'un actionneur de type manette analogique à 2 axes : *joystick*...).

Procédure type d'utilisation lorsque le robot est en évolution :

- le robot est en pilotage mode manuel, il est localisé au point P1 sur le site à explorer, sa position est montrée à l'opérateur sur une carte du site ;
- Depuis ce mode manuel, l'opérateur indique au robot la nouvelle position à atteindre P2 en lui transmettant les coordonnées adéquates ;
- le robot passe en mode automatique et se dirige vers ce point.
- *le robot transmet sa position géographique (1) périodiquement, qu'il soit en mode manuel ou automatique ;*
- si nécessaire, l'opérateur peut en mode automatique modifier la vitesse d'évolution du robot en fixant un pourcentage de la vitesse nominale pré-réglée (ce pourcentage est par défaut égal à 50) ;
- lorsque un obstacle infranchissable se présente devant le robot, celui-ci le détecte (les capteurs utilisés pour cela ne sont pas concernés par la présente étude) et s'arrête. Il transmet alors au système distant un signal « obstacle détecté » ;
- Le mode arrêt permet à l'opérateur de reprendre le contrôle manuel du robot afin de lui faire contourner l'obstacle. Il est aidé pour cela par les images caméra ;
- *les consignes sont envoyées sous forme de couples direction/vitesse (2), elles sont traduites par le robot qui calcule les commandes de vitesse des roues, indépendante pour chacune d'elles ;*
- lorsque l'opérateur considère que la voie est à nouveau libre, il peut demander au robot de reprendre sa trajectoire en mode automatique vers le point P2 ;
- lorsque la position P2 est atteinte, le robot s'arrête et signale à l'opérateur l'évènement « position atteinte ». Ce dernier peut reprendre la main manuellement pour explorer localement le site, prélever d'éventuels échantillons, ou spécifier une nouvelle position à atteindre en automatique ;
- *l'opérateur peut à tout moment émettre l'ordre d'arrêt (3) du robot quel que soit son mode de fonctionnement.*

Le diagramme UML du document réponse (C.2.2) va être utilisé pour modéliser les différentes possibilités de mobilité du RoboVolc...

Question C.2.1.

Comment s'appelle le type de diagramme du document réponse C.2.2. dans la terminologie UML ?
Quel est son rôle ?

La phase d'initialisation, le mode arrêt du robot (4), et les *affirmations en italique* du texte précédent sont déjà modélisés sur le diagramme. Le signal (5) montre un exemple d'évènement avec déclenchement d'action (évènement / action).

Question C.2.2.

Après lecture attentive de la procédure type précédente, compléter le diagramme en faisant apparaître les **modes** « pilotage manuel » et « mode automatique », les **événements** et les **actions** de manière à couvrir tous les extraits soulignés dans le texte.

C.3. Centralisation et gestion des données

Les questions suivantes sont relatives à la gestion des trajectoires du robot sur le poste de supervision. L'ensemble des données manipulées est stocké en base de données. Une étude préalable a permis d'expliciter les termes suivants.

- Une campagne d'exploration est définie par une date de début et une durée prévisionnelle, une description du site, et des renseignements divers concernant l'objet de l'exploration, les chercheurs impliqués, le nom du manipulateur, la météo...
Une campagne peut se traduire par plusieurs jours de travail. Chaque journée peut faire l'objet de plusieurs trajectoires et plusieurs prélèvements.
- Une trajectoire est définie par une succession de points de passage et par la distance totale parcourue. Le nombre de points n'étant pas limité, ils seront stockés dans une liste chaînée (en pratique, on utilisera la classe template `list` proposée par la STL).
Lorsque le robot est piloté en manuel, la partie de trajectoire réalisée est modélisée de proche en proche sous forme d'une succession de points de passages.
Une trajectoire se termine en fin de campagne ou lorsque l'opérateur en initialise une nouvelle.
- Un prélèvement est défini par un numéro d'ordre et par le point géographique où il a eu lieu. Il peut être de type prélèvement de gaz ou prélèvement de minéraux.
Le robot peut stocker un seul prélèvement gazeux et possède 2 paniers distincts pour les prélèvements minéraux, la capacité de chaque panier est de 1000 cm³.
- Un point de passage est caractérisé par ses coordonnées longitude-latitude-altitude, et une datation UTC (Temps Universel). Une information précise s'il s'agit d'un point programmé ou d'un point calculé suite à un pilotage manuel. Un point peut le cas échéant être référencé par un prélèvement qui y a été effectué.
Un point programmé comme objectif, sera systématiquement mis à jour lorsque le robot l'aura atteint, ce sont alors les coordonnées fournis par le GPS qui permettent de connaître l'altitude...

Question C.3.1.

Faire apparaître sur le diagramme de classes fourni dans le document réponse les relations entre les différentes classes, et les multiplicités de ces relations.

Question C.3.2.

Proposer une déclaration en langage C++ de la classe Trajectoire restreinte à la description ci-dessus et au diagramme de classe.

Une partie des données est stockée en local sur le robot, c'est notamment le cas des trajectoires de la journée, des points de passages et des prélèvements. Le moteur de cette base de données est SQLite3. Ce SGBD-R est conforme à la norme SQL 92.

Le modèle conceptuel de données (MCD) est global au système. La présente étude porte uniquement sur les points de passages et les prélèvements. Ci-après le MCD correspondant :

PointDePassage	
idPDP	INTEGER
automatique	BOOL
date	DATETIME
longituede	REAL
longituedeEW	CHAR
latitude	REAL
latitudeNS	CHAR
altitude	REAL

Prelevement	
idPrev	INTEGER
mineraux	BOOL
idPDP	INTEGER
numPanier	INTEGER

Il est à noter que les attributs `longituedeEW` et `latitudeNS` peuvent prendre respectivement 'E' ou 'W' et 'N' ou 'S' comme valeurs. Ceci correspond au codage des longitudes et latitudes (un nombre réel + une direction).

```
SELECT longitude, longituedeEW,
        latitude, latitudeNS,
        altitude, date
FROM PointDePassage
WHERE automatique = TRUE ;
```

La requête présentée permet de connaître tous les points de passage automatique ainsi que la date et l'heure de ce passage.

Question C.3.3.

En vous appuyant sur l'exemple ci-dessus, donner la requête SQL permettant de connaître le numéro de tous les prélèvements gazeux.

```
SELECT longitude, longituedeEW,
        latitude, latitudeNS,
        altitude, date
FROM PointDePassage, Prelevement
WHERE PointDePassage.idPDP = Prelevement.idPDP
      AND mineraux = FALSE
      AND altitude >= 500 AND altitude < 501 ;
```

Attention. Jointure indispensable puisque `mineraux` appartient à `Prélèvement` et `altitude` à `PointDePassage`.

La requête présentée permet de connaître les lieux de tous les prélèvements gazeux qui ont eu lieu à 500m d'altitude. Attention, l'altitude étant un REAL nécessite un intervalle, ici [500 ; 501[mètres.

Question C.3.4.

Donner la requête SQL permettant de connaître le numéro des prélèvements, le type de prélèvements ainsi que le numéro de panier du point de coordonnées 4807.048, N, 02131.324, E à 545 m d'altitude.

D. Communication et réseau

D.1. Protocoles

Le réseau mis en œuvre entre le robot et le poste de supervision est de type WIFI 802.11g. Ce type de liaison radio peut s'assimiler pour le cas présent à une connexion Ethernet très peu fiable (perturbations dues à la conformation du terrain et à l'électromagnétisme).

Les données à transmettre sont des ordres pour donner une trajectoire ou pour piloter le robot en mode manuel. Ces ordres sont transmis via une transmission UDP.

Question D.1.1.

Quelle information des protocoles TCP et UDP identifie sans équivoque le processus destinataire du message ?

Cette information peut prendre des valeurs dont certaines sont réservées par la DARPA pour des services usuels. Le fichier `/etc/services` recense l'ensemble de ces services.

Question D.1.2.

Citer les trois informations définies par ce fichier pour chacun des services.

Question D.1.3.

Un même numéro de port peut-il être utilisé simultanément en TCP et en UDP par deux processus distincts ? Justifier votre réponse.

Question D.1.4.

Quels sont les particularités respectives des protocoles TCP et UDP ? Compléter le document réponse.

D.2. Sécurité des biens et des personnes

La communication WIFI étant très peu fiable, un mécanisme de détection de rupture de flux est mis en place.

Question D.2.1.

D'après le diagramme de déploiement, quelle solution matérielle peut être utilisée pour transmettre les informations de commande lors d'une rupture de flux WIFI ?

Question D.2.2.

Quelle action doit être envisagée en cas de perte totale du flux de données ?

L'équipe de RoboVolc accepte que le robot se déplace sur 0,25 m sans communication avec le superviseur.

6 26/26

La détection de rupture de flux est réalisée par le mécanisme suivant :

```
// coté robot
N <- valeur aléatoire
ERREUR <- 0
FAIRE
  Envoyer N au robot // via protocole UDP sur le port 3333
  Attendre la réception d'un entier R du robot pendant X ms
  SI pas de réception dans le délai imparti
    ERREUR <- ERREUR + 1
  SINON
    SI ( R == N )
      ERREUR <- 0
      N <- N + 1
    SINON
      ERREUR <- ERREUR + 1
    FIN SI
  FIN SI
  SI ( ERREUR >= 5 )
    Appeler procédure Robot::enCasDeRuptureDeFlux ()
    N <- valeur aléatoire
  FIN SI
TANT QUE ( VRAI )
```

```
// coté poste de supervision
MEM <- 0
ERREUR <- 0
FAIRE
  Attendre la réception d'un entier R du superviseur pendant X ms
  SI pas de réception dans le délai imparti
    ERREUR <- ERREUR + 1
  SINON
    SI ( ( MEM + 1 ) == R )
      ERREUR <- 0
    SINON
      ERREUR <- ERREUR + 1
    FINSI
    MEM <- R
    Envoyer R au superviseur
  FIN SI
  SI ( ERREUR >= 5 )
    Appeler procédure Supervision::enCasDeRuptureDeFlux ()
  FIN SI
TANT QUE ( VRAI )
```

Remarque : Cet algorithme sera implémenté à la question D.3.2.

Question D.2.3.

Au bout de combien de tentatives infructueuses de réception de données les procédures enCasDeRuptureDeFlux () sont-elles appelées ?

Question D.2.4.

Calculer en millisecondes (ms) la valeur de X de l'algorithme précédent de manière à respecter les contraintes citées pour les deux vitesses suivantes du robot :

- 1 m/s ?
- 5 m/s ?

D.3. Programmation réseau

Soit la classe `DatagramSocket` permettant d'envoyer et de recevoir des paquets de données via le protocole UDP :

```
class DatagramSocket
{
private:
    unsigned short port ;
    int sock ;
    struct sockaddr_in source ;

public:
    DatagramSocket( unsigned short port = 0 ) ;
    ~DatagramSocket() ;

    // fixe le timeout en ms

    void setTimeout( unsigned int timeout ) ;

    // La méthode readDatagram() attend la réception d'un datagramme
    //
    // Le résultat est stocké dans data, maxlen est la taille de data
    //
    // Valeur renvoyée : la longueur du datagramme lu,
    //                   ou -1 en cas d'erreur,
    //                   ou -2 en cas de timeout

    long readDatagram( void *data, long maxlen ) ;

    long writeDatagram(
        const void *data,      // données à émettre
        long len,             // taille des données à émettre
        const char* host,     // machine cible
        unsigned short port   // port destinataire
    ) ;
};
```

Fonction de lecture d'un datagramme.

Fonction d'écriture d'un datagramme.

Le constructeur de `DatagramSocket` prend en argument le numéro du port local, le sauvegarde, créer une socket et l'attache si nécessaire (`port > 0`).

Question D.3.1.

A l'aide de l'annexe 4, proposer une implémentation du constructeur

```
DatagramSocket( unsigned short port = 0 ) ;
```

Au bout de la durée X ms fixée par `setTimeout()`, la méthode `readDatagram()` renvoie `-2` si aucun datagramme n'a été reçu.

Question D.3.2.

Compléter, d'après le pseudo-code des pages précédentes, l'implémentation de la méthode

```
void Superviseur::keepAlive(
    DatagramSocket *ds, // Instance de Datagram pour communication
    char* robotName,    // Nom (hostname) du destinataire
    int port );        // Port à utiliser
```

permettant la détection de rupture de flux coté poste de supervision

On considère les modifications suivantes de la méthode `readDatagram()` :

```
// retourne : nombre d'octet lues
// exception: transmet une exception TimeoutException
//           au bout d'un timeout de X ms
//           défini par la méthode setTimeout(X)
long readDatagram(
    char *data,
    long maxlen )
    throw (TimeoutException) ;
```

Voir annexe 5 « les exceptions en C++ » et l'exemple de mise en oeuvre qui y figure.

Question D.3.3.

Si la fonction membre `readDatagram()` de la classe `DatagramSocket` avait le prototype modifié proposé ci-dessus, comment pourrait-on intercepter l'émission de l'exception liée au timeout lancée par la méthode ?

D.4. Mise au point

En phase de mise au point des algorithmes précédents, le robot ouvre sa socket de communication sur le port 4444, tandis que le superviseur ouvre la sienne sur le port 5555. Les trois trames suivantes ont alors été capturées sur le réseau Wifi entre le superviseur et le robot :

Soit trois trames « décodées par Ethereal » :

● Trame 1

```
⊕ Frame 11 (42 bytes on wire, 42 bytes captured)
⊕ Ethernet II, Src: 192.168.0.16 (00:0f:b0:71:c4:c1), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊖ Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  Sender MAC address: 192.168.0.16 (00:0f:b0:71:c4:c1)
  Sender IP address: 192.168.0.16 (192.168.0.16)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.0.219 (192.168.0.219)
```

● Trame 2

```
⊕ Frame 12 (64 bytes on wire, 64 bytes captured)
⊕ Ethernet II, Src: 192.168.0.219 (00:04:00:dc:1c:2d), Dst: 192.168.0.16 (00:0f:b0:71:c4:c1)
⊖ Address Resolution Protocol (reply)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (0x0002)
  Sender MAC address: 192.168.0.219 (00:04:00:dc:1c:2d)
  Sender IP address: 192.168.0.219 (192.168.0.219)
  Target MAC address: 192.168.0.16 (00:0f:b0:71:c4:c1)
  Target IP address: 192.168.0.16 (192.168.0.16)
```


● Trame 3

```
⊕ Frame 6 (44 bytes on wire, 44 bytes captured)
⊕ Ethernet II, Src: 192.168.0.16 (00:0f:b0:71:c4:c1), Dst: 192.168.0.219 (00:04:00:dc:1c:2d)
⊕ Internet Protocol, Src: 192.168.0.16 (192.168.0.16), Dst: 192.168.0.219 (192.168.0.219)
⊕ User Datagram Protocol, Src Port: 4444 (4444), Dst Port: 5555 (5555)
    Source port: 4444 (4444)
    Destination port: 5555 (5555)
    Length: 10
    Checksum: 0x89df [correct]
    Data (2 bytes): cc af
```

L'étude du décodage des trames précédents met en évidence l'utilisation du protocole ARP.

Question D.4.1.

Compléter le tableau du document réponse en indiquant si les propositions relatives au protocole ARP sont vraies ou fausses.

Question D.4.2.

Après avoir analysé le contenu de la trame 3, compléter le tableau du document réponse.

Question D.4.3.

Ce datagramme (trame 3) est-il émis du robot vers le poste de supervision ou du poste de supervision vers le robot ? Justifier.

Fin du questionnaire.