

BTS INFORMATIQUE ET RESEAUX

POUR L'INDUSTRIE ET LES SERVICES TECHNIQUES

Session 2007

Épreuve E.4

Étude d'un Système Informatisé

Prélèvements sur sites volcaniques

Éléments de correction

et proposition de barème détaillé (sur 100 points)

(13 pages)

Mise en page identique au document réponse

B. Le robot mobile

B.1.1. Compléter le tableau avec les symboles

(2 points)

roue	G1	G2	G3	D1	D2	D3
avance A..B	+	+	+	+	+	+
recule B..C	-	-	-	-	-	-
tourne sur lui-même en C	- - 0	- - 0	- - 0	0 + +	0 + +	0 + +
avance C..D	+	+	+	+	+	+
arrêt en D	0	0	0	0	0	0

Il y a plusieurs solutions pour faire tourner le robot à gauche, la 2^{ème} semble la meilleure pour qu'il tourne sur lui-même...

B.1.2. Exprimer Ω_R en fonction de D et de V_D . (2 points)

$$\Omega_R = 2 \cdot V_D / D \quad (\text{en rad/s})$$

$$\Omega_R = V_D \cdot 60 / (\pi \cdot D) \quad (\text{en tr/min}) \quad \text{ou bien } N = \Omega_R \cdot 60 / (2 \cdot \pi)$$

Puis faire l'application numérique.

$$\Omega_R \text{ en rad/s} = 1,33 \text{ rad/s}$$

$$\Omega_R \text{ en trs/min} = 12,73 \text{ trs/min}$$

B.1.3. Valeur numérique de U(t) pour obtenir une valeur nulle en sortie du CNA ? (2 points)

$$U(t) = (U - V_{Amin}) \cdot (2^N - 1) / (V_{Amax} - V_{Amin}) \quad (2 \text{ points})$$

avec $V_{Amax} = +10$ et $V_{Amin} = -10$

si $U = 0$, alors $U(t) = 32768$ (on peut accepter un résultat intuitif sans calcul)

B.1.4. Consigne U correspondant à la valeur d'entrée U(t) = 0xA000 ? (2 points)

$$U = (V_{Amax} - V_{Amin}) / (2^N - 1) \cdot U(t) + V_{Amin} \quad (2 \text{ points})$$

avec $U(t) = 0xA000 = 40960$

$$U = 20 / 65535 \cdot 40960 - 10 = + 2,50 \text{ V} \quad (\text{résultat intuitif sans calcul accepté})$$

B.1.5. Indiquer les combinaisons valides de choix de plage et de gain cohérentes avec la tension retournée par la dynamo tachymétrique.

Répondre par O (Oui) ou N (Non) (2 points)

plage	0..10V			$\pm 5V$			$\pm 10V$		
gain	x 1	x 10	x 100	x 1	x 10	x 100	x 1	x 10	x 100
O/N	N	N	N	O	N	N	O	O	N

B.1.6. Parmi les configurations valides ci-dessus, quelle est celle qui semble la mieux adaptée au problème ? (2 points)

Plage : $\pm 10V$ (2 points)

Gain : x 10 (63% de l'étendue de mesure)

B.1.7. Calculer la résolution Res en mV du CAN pour la configuration retenue. (1 point)

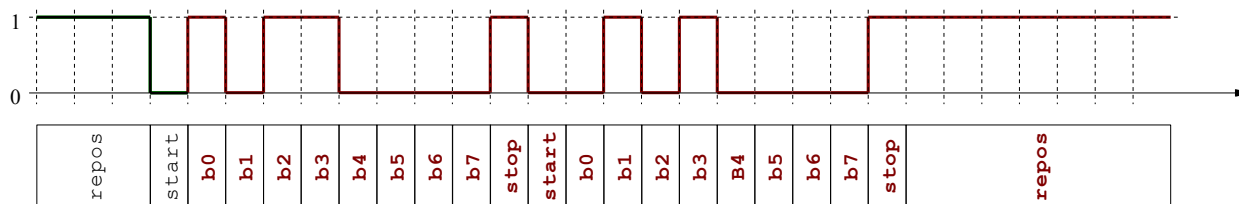
$$\text{résolution} = \text{étendue_de_mesure} / 2^N \quad (1 \text{ point})$$

$$\text{Res} = 4,882 \text{ mV} \quad (\text{ou la moitié si la plage choisie en B.1.6. est } \pm 5V)$$

B.1.8. Quel(s) autre(s) type(s) de capteur peut être employé pour mesurer une vitesse de rotation ? (1 point)

Codeur incrémental (ou autre solution plausible proposée)

B.2.1. Chronogramme de transmission des caractères CR et LF (4 points)



Valeur de M_t ?

$M_t = 1 / 4800 \text{ secondes}$, $M_t \approx 210 \mu s$

B.2.2. Configuration de l'UART pour communiquer avec le GPS. (4 points)

		Nom du registre	adresse relative	valeur hexa.
1	Mise à 1 du bit DLAB	LCR	3	0x80
2	Poids faible du diviseur	DLLB	0	0x18
3	Poids fort du diviseur	DLHB	1	0x00
4	Sélection du protocole, et DLAB à 0	LCR	3	0x03

En 1, seul b7 est significatif... 128 réponses possibles...

B.2.3. Prototype de la fonction sommeDeControle() (1 point)

```
unsigned char  sommeDeControle( char* trame ) ;
```

B.2.4. Implémentation de la fonction sommeDeControle() (4 points)

```
unsigned char  sommeDeControle( char* trame )
{
    unsigned char  result = 0 ;
    while ( *++trame != '$' )      result ^= *trame ;
    return result ;
}
```

Il s'agit d'une solution très optimisée... La pré-incrémentation permet de sauter le '\$' de début de trame. Ne pénaliser aucune petite erreur (; oublié par exemple) . Le but est de voir apparaître une boucle, avec une borne d'arrêt cohérente, un ou exclusif et une manipulation pointeur, valeur pointée et incrémentation...

En C, la différence des types char / unsigned char ne conduira pas à un avertissement...

B.2.5. Exprimer en degrés, minutes, secondes les coordonnées du robot.

2114.35,S : **21° 14' (0.35 x 60)" = 21° 14' 21" Sud** **(1 point)**

05540.305,E : **55° 40' (0.305 x 60)" = 55° 40' 18" Est**

(c'est à dire près d'un célèbre volcan sur l'île de la Réunion...)

B.2.6. Calculer le temps de transmission d'une trame constituée de N caractères.

1 caractère = 1 start + 8 data + 1 stop = 10 bits **(1 point)**

T = N . 10 / 4800 secondes = N / 480 secondes

B.2.7. En déduire le temps de transmission de la trame GLL précédente

longueur de la trame = 43 (sans oublier CR et LF) **(1 point)**

T = 43 / 480 = 0,0896 s = 89,6 ms

B.3.1. Quel est le risque encouru par l'application ? **(1 point)**

Obsolescence ou incohérence d'une partie des données .

Toute réponse qui met en évidence ce concept ou ses conséquences doit être acceptée.

B.3.2. Qu'appelle-t-on une section critique dans un système multitâche ?

Zone de code non interruptible. **(1 point)**

B.3.3. Nom des fonctions permettant de réaliser les opérations suivantes ?

(3 points)

Créer un sémaphore	<code>rtf_sem_init()</code>
Prendre un sémaphore	<code>rtf_sem_wait()</code>
Rendre un sémaphore	<code>rtf_sem_post()</code>

B.3.4. Implémentation de la fonction initSem() .

(3 points)

```
bool initSem ( int numSem )
{
    return !rtf_sem_init( numSem, 1) ;
}
```

Ne pas pénaliser outre mesure une inversion du résultat...

B.3.5. Insérer le code permettant de rendre atomique l'écriture des 6 vitesses dans

la mémoire partagée.

(2 points)

```
void envoyerConsignes(ShmMemWrite* shm, // Zone mémoire
                      float* cons,      // Table des consignes
                      int numSem )      // Id. du sémaphore
{
    /*    ... Code avant écriture dans la shm ...    */

    Réponse : rtf_sem_wait( numSem ) ;
               for ( i = 0 ; i < 6 ; i++ ) shm->v[ i ] = cons[ i ] ;

    Réponse : rtf_sem_post( numSem ) ;
               /*    ... Code après écriture dans la shm ...    */
               return;
}
```

C. Contrôle-commande et supervision.

C.1.1. Que signifie la flèche reliant Opérateur de maintenance à Opérateur ?
(2 points)

Opérateur de maintenance dérive de (est une spécialisation de) Opérateur

C.1.2. Pour chaque diagramme, indiquer dans le tableau récapitulatif si il est correct ou non. Justifier en cas de réponse négative.
(4 points)

Diagramme n°1

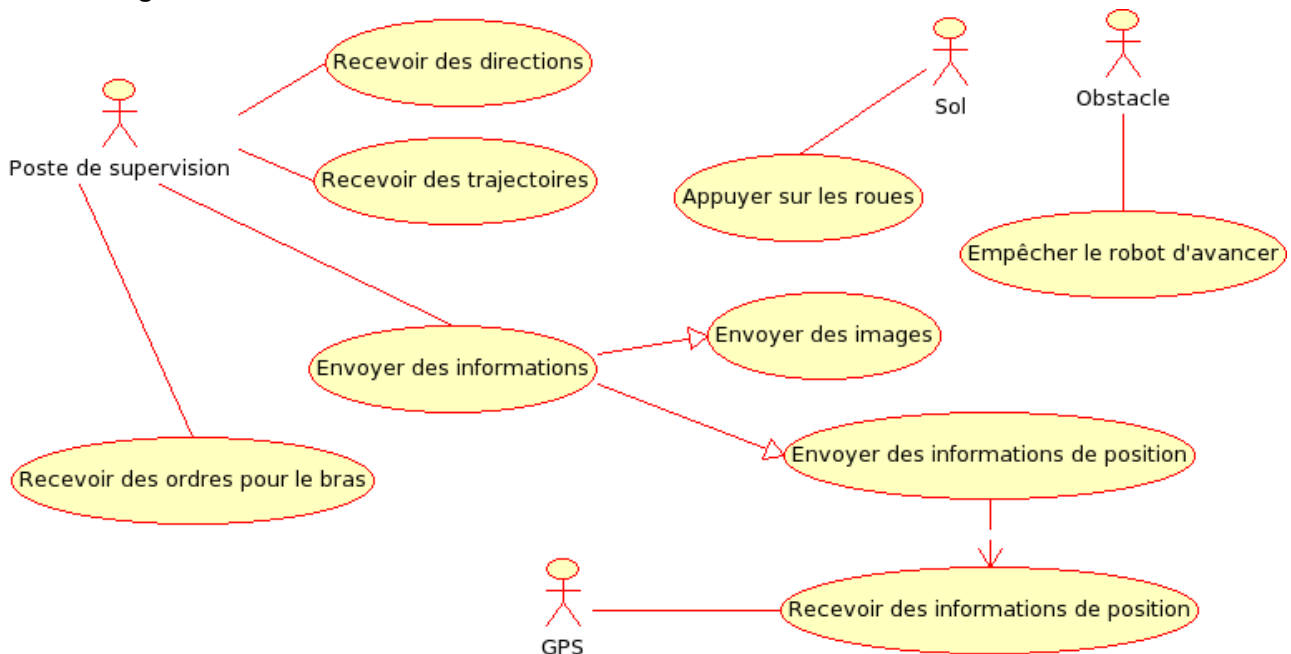


Diagramme n°2

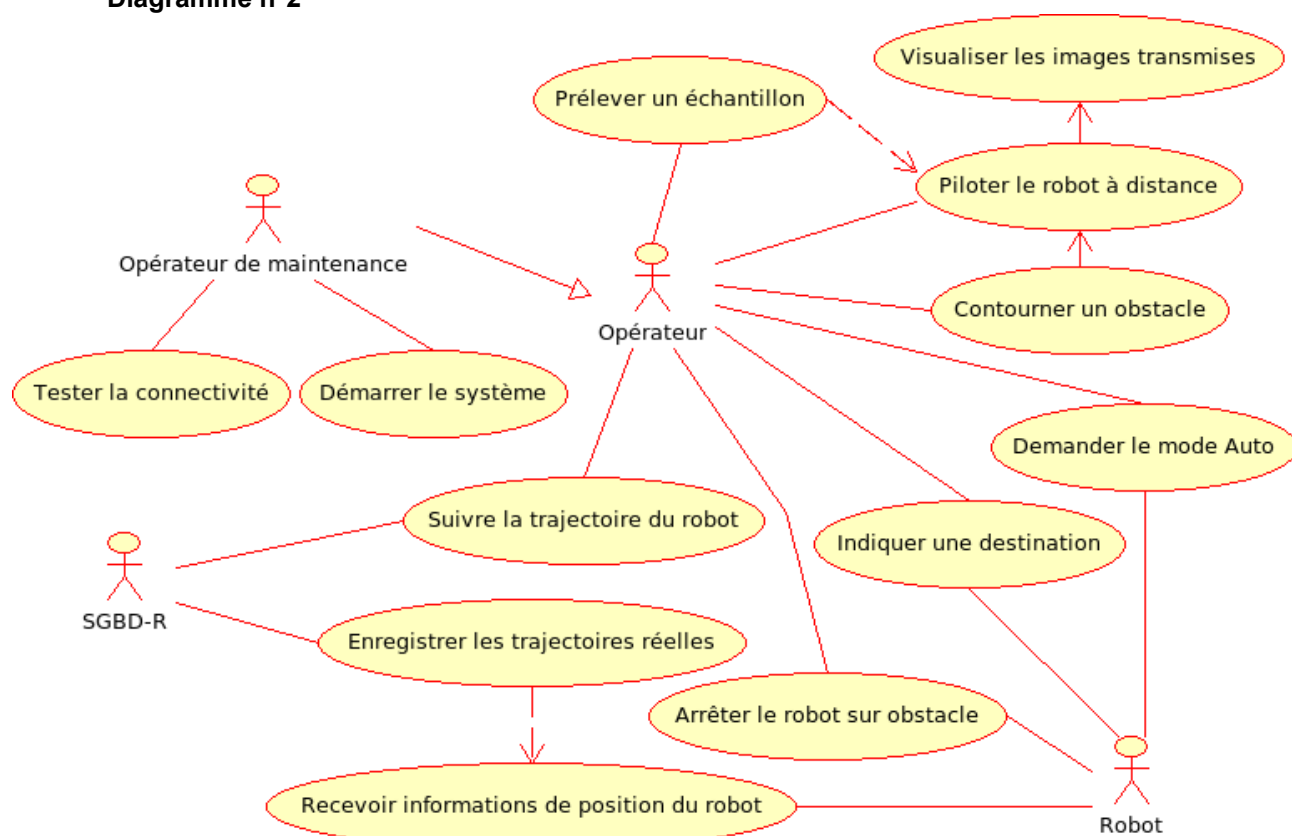


Diagramme n°3

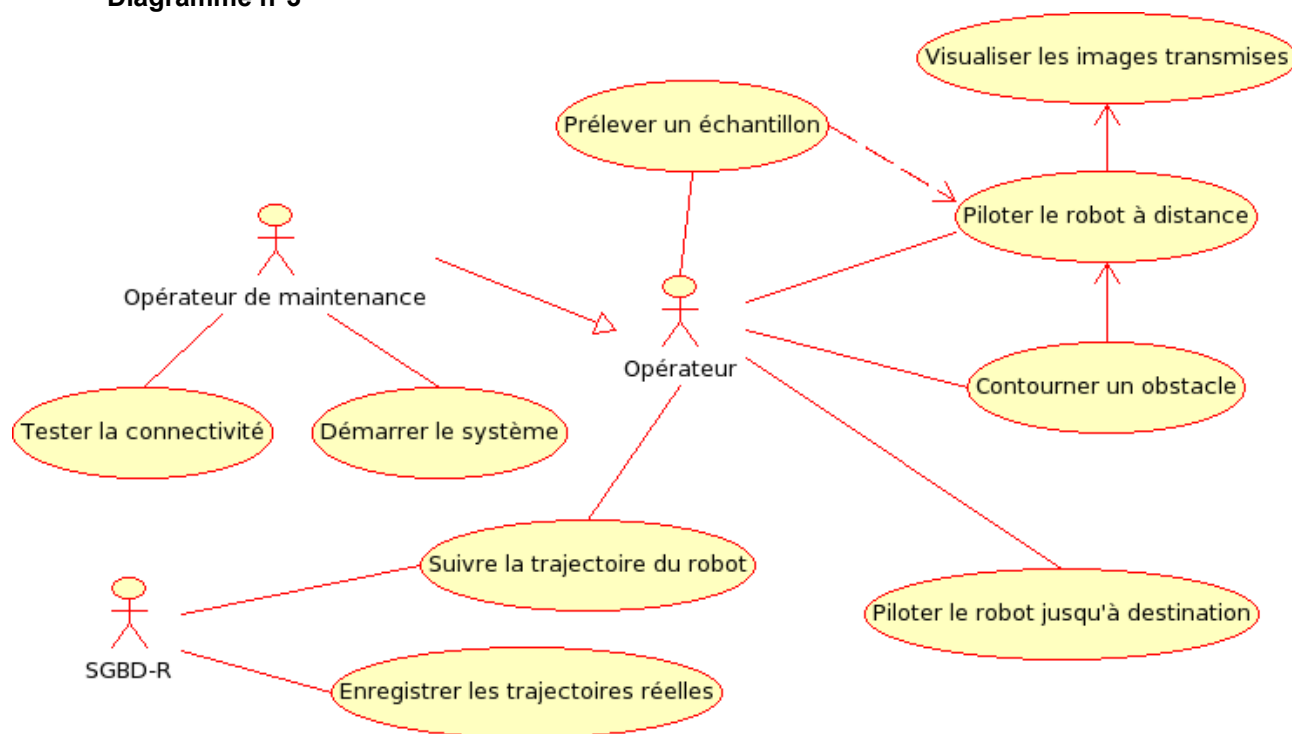


Diagramme n°4

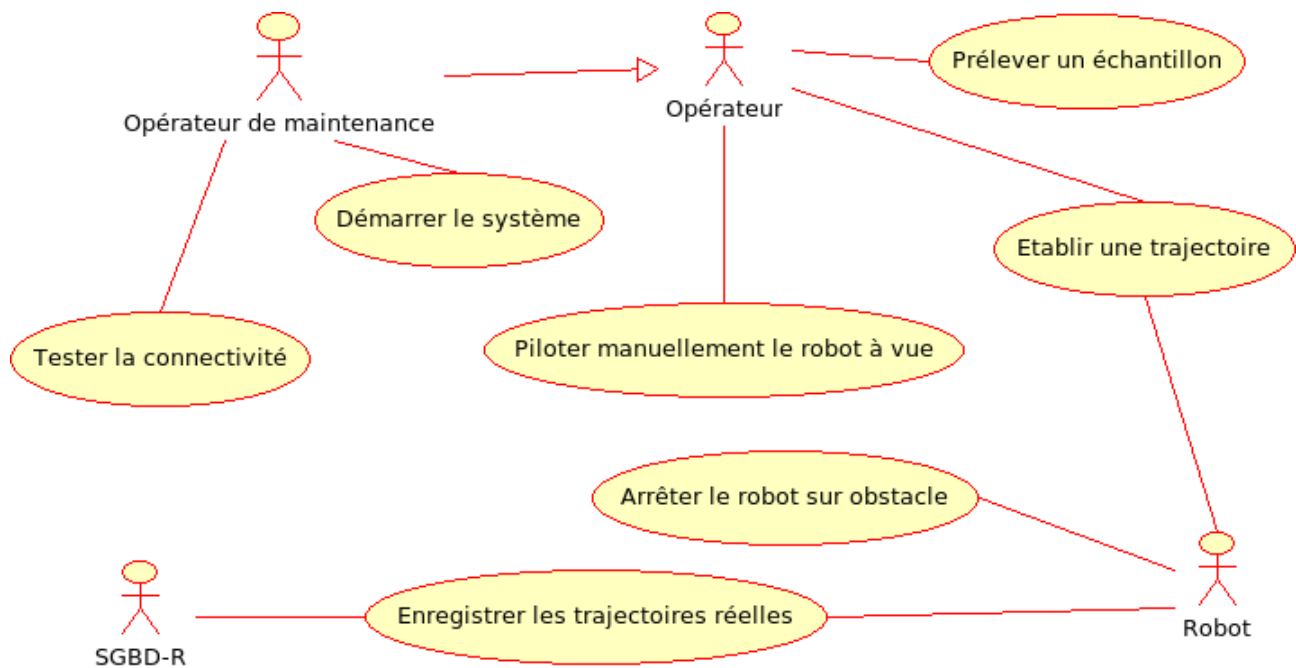


Tableau récapitulatif:

Diag.	Correct (oui/non)	Justification si non
n° 1	NON	point de vue du robot
n° 2	OUI	
n° 3	NON	acteur Robot absent
n° 4	OUI	

(toute justification prouvant qu'un diagramme est faux peut être acceptée...)

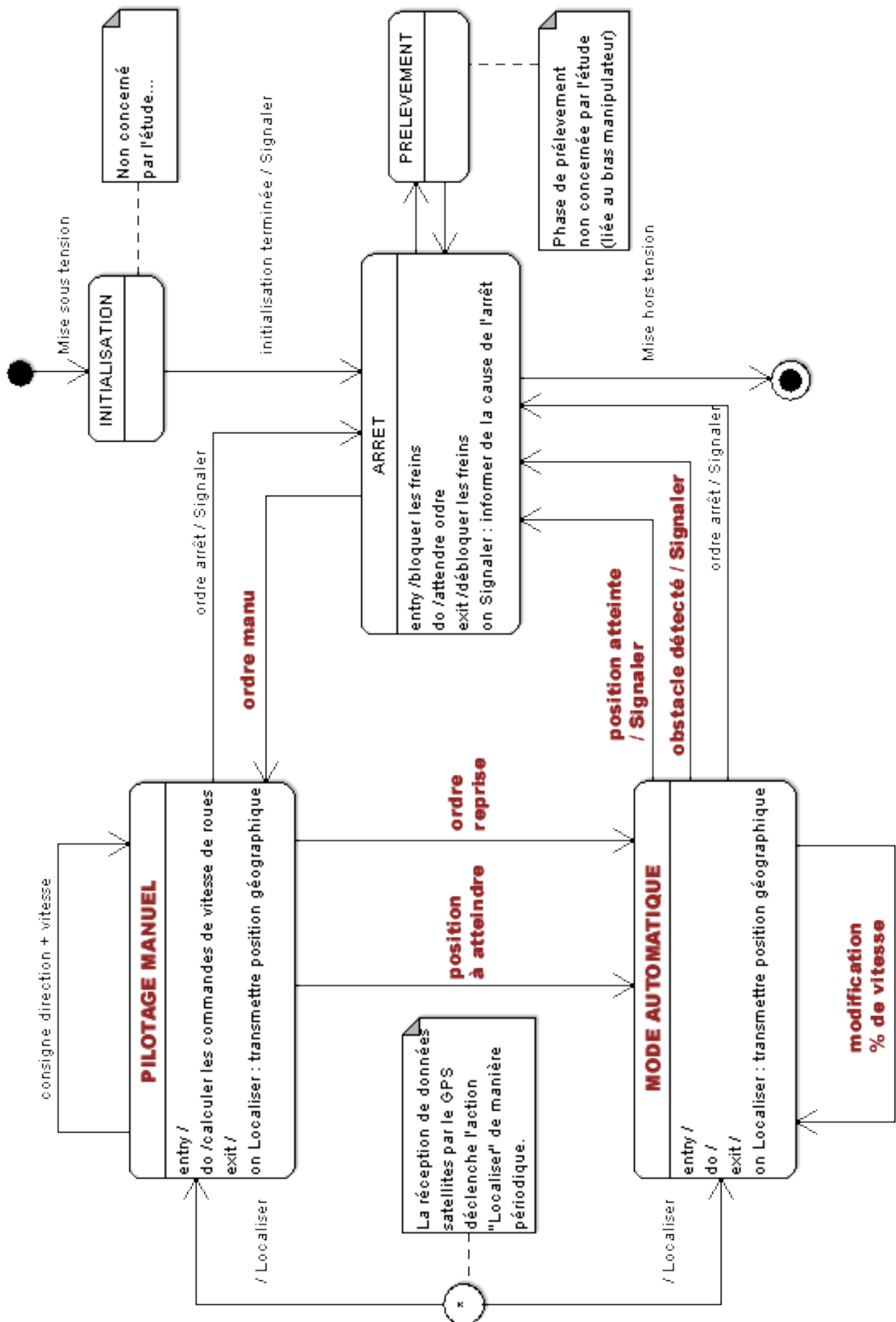
C.2.1. Type et rôle du diagramme UML de la question C.2.2 ? (2 points)

Type : Diagramme États – Transitions

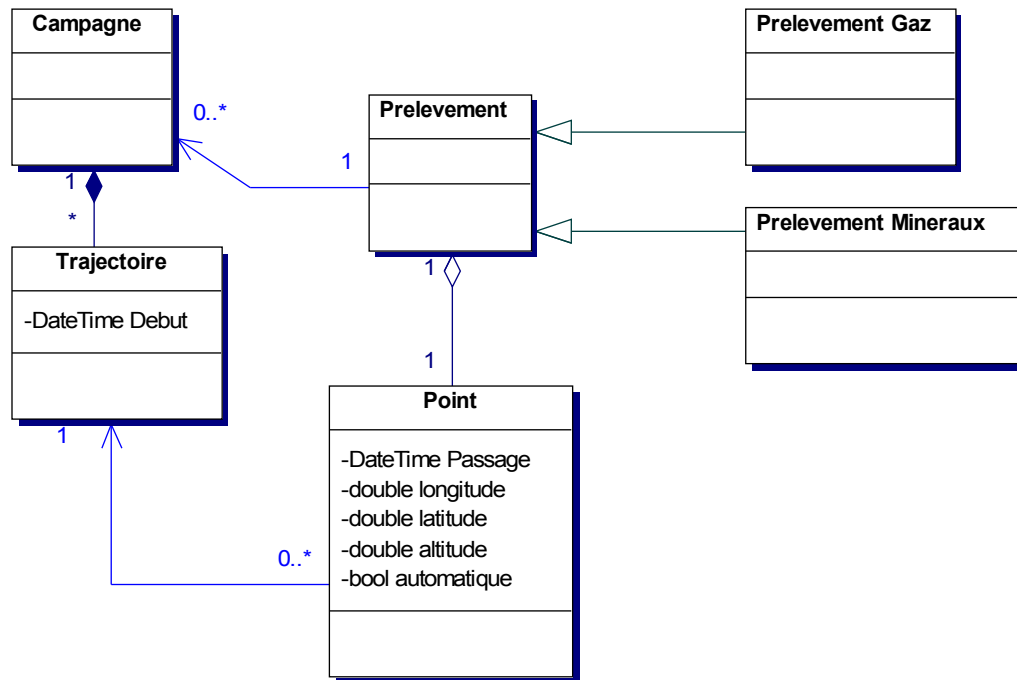
Rôle : montrer le comportement dynamique d'un objet ou d'un système

C.2.2. Compléter le diagramme.

(6 points)



C.3.1. Faire apparaître sur le diagramme de classes les relations entre les différentes classes, et les multiplicités de ces relations. (3 points)



C.3.2. Proposer une déclaration en langage C++ de la classe Trajectoire. (4 points)

```

typedef list<Point> LISTPOINT;

class Trajectoire
{
private:
    LISTPOINT ListeDesPoints;

public:
    Trajectoire(DateTime Debut, Point &P);
    ~Trajectoire();
    unsigned int DistanceTotaleParcourue();
    void AjouterPointDePassage(Point &P);
};
  
```

La distance totale parcourue pourrait également être un attribut public.

Le constructeur peut ne pas prendre de paramètre. (DateTime remplie par une API, et le premier point fixant l'origine de la liste).

La méthode AjouterPointDePassage() peut être présentée différemment...

Sans que cela soit pour autant une question ouverte, on peut envisager ici de nombreuses réponses. On s'attachera à juger la logique et la faisabilité de la solution proposée.

C.3.3. Donner la requête SQL permettant de connaître le numéro de tous les prélèvements gazeux. (2 points)

```

SELECT idPrev
FROM Prelevement
WHERE mineraux = FALSE ;
  
```

C.3.4. Donner la requête SQL permettant de connaître le numéro des prélèvements, le type de prélèvement ainsi que le numéro de panier du point de coordonnées 4807.048,N ; 02131.324,E à 545m d'altitude. (2 points)

```
SELECT idPrev , numPanier , mineraux
FROM PointDePassage , Prelevement
WHERE PointDePassage.idPDP = Prelevement.idPDP
AND altitude > 544 AND altitude < 546
AND longitude = 02131.324 AND longitudeEW = 'E'
AND latitude = 4807.048 AND latitudeNS = 'N' ;
```

D. Communication et réseau

D.1.1. Quelle information des protocoles TCP et UDP identifie sans équivoque le processus destinataire du message ? (2 points)

le numéro de port

D.1.2. Citer les trois informations définies par /etc/services pour chacun des services. (3 points)

- 1) nom du service
- 2) numéro du port
- 3) protocole (tcp ou udp)

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. (2 points)

oui,

car ce sont des numéros logiques avec des espaces d'adressage propres

D.1.4. Particularités des protocoles TCP et UDP ? (3 points)

	TCP	UDP
numéro et nom de couche OSI	4, transport	4, transport
protocole sous-jacent	IP	IP
fiabilité du transport	OUI	NON
séquencement des données	OUI (automatique)	NON (manuel)
mode de connexion	Connecté	Non connecté
taille maximale des données	Infinie	Taille MAX du prot. IP
possibilité de diffusion	NON	OUI

D.2.1. Quelle solution matérielle peut être utilisée pour transmettre les informations de commande lors d'une rupture du flux WiFi ? (1 point)

Le modem radio

D.2.2. Quelle action doit être envisagée en cas de perte totale du flux de données ?

Arrêter le robot, (2 points)
voire reculer si la température ambiante est trop élevée...

D.2.3. Au bout de combien de tentatives infructueuses de réception de données les procédures enCasDeRuptureDeFlux() sont-elles appelées ? (1 point)

5

D.2.4. Calculer en millisecondes la valeur de X dans les cas suivants de vitesse du robot ? (2 points)

1 m/s : 50 ms **temps = distance / (vitesse . nb_tentatives)**
5 m/s : 10 ms

D.3.1. Implémentation du constructeur DatagramSocket(unsigned short port = 0) ; (5 points)

```
DatagramSocket::DatagramSocket( unsigned short port )
{
    this->port = port ;

    sock = socket( PF_INET, SOCK_DGRAM, IPPROTO_UDP ) ;
    memset( &source, 0, sizeof( source ) ) ;
    source.sin_family = AF_INET ;
    source.sin_addr.s_addr = htonl( INADDR_ANY ) ;
    source.sin_port = htons( port ) ;
    if ( port > 0 ) bind( sock,
                        (struct sockaddr *) &source,
                        sizeof( source ) ) ;
}
( version minimale... )
```

D.3.2. Compléter l'implémentation de la méthode `Superviseur::keepAlive()` permettant la détection de rupture de flux coté poste de supervision.

(4 points)

```
void Superviseur::keepAlive (Datagram* ds, char* robotName, int port)
{
    unsigned short r ;
    unsigned short mem = 0;
    int erreur = 0 ;
    do {
        if( ds->readDatagram( (void *)&r , sizeof(r) ) < (int)sizeof(r) ){
            erreur ++ ;    // pas de réception dans le délai imparti
        } else {
            // réception...
            if ( mem == ( r - 1 ) ) {
                erreur = 0 ;
            } else {
                // pas de concordance
                erreur++ ;    // avec les envois préc.
            }
            mem = r ;

            long res = ds->writeDatagram( (void*)&r,
                                          sizeof(r),
                                          "robot", 4444 ) ;

            if ( res < (int)sizeof(r) ) {
                erreur++ ;
            }
        }
        if ( erreur >= 5 ) {
            Supervision::enCasDeRuptureDeFlux() ;
        }
    } while ( true ) ;
}
```

D.3.3. Interception de l'exception liée au timeout, lancée par la méthode `readDatagram()` ?

(3 points)

```
try {
    readDatagram( ... ) ;
}

catch ( TimeoutException e ) {

    // traitement de la non réponse
}
```

Le mécanisme des exceptions étant complètement revu en annexe, on peut attendre du candidat une réponse juste...

D.4.1. Protocole ARP.**(3 points)**

Vrai / Faux	Propositions
Faux	ARP est un protocole de couche 6.
Vrai	ARP signifie protocole de résolution d'adresses.
Vrai	Une requête ARP est forcément en diffusion.
Faux	ARP est utilisé par un ordinateur lorsqu'il souhaite émettre une trame Ethernet à une autre machine dont il ne connaît que l'adresse MAC.
Vrai	Si l'adresse IP est présente dans le cache de l'émetteur, il suffit de lire l'adresse MAC correspondante pour envoyer la trame Ethernet.

D.4.2. Caractéristiques de la trame n°3 .**(3 points)**

Questions	Réponses
Adresse MAC de l'émetteur	00:0F:B0:71:C4:C1
Adresse MAC du destinataire	00:04:00:DC:1C:2D
Adresse IP de l'émetteur	192.168.0.16
Adresse IP du destinataire	192.168.0.219
Port Source	4444
Port Destination	5555
Données en décimal	52399 (1 donnée 16 bits)

D.4.3. Sens de circulation de la trame n°3 ? Justifier.**(1 point)****Du robot vers la Supervision****(le port source 4444 est celui du robot)**