MINISTÈRE
DE L'ÉDUCATION
NATIONALE

EDE NUM 2

SESSION 2018

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.

Dans le cas où un(e) candidat(e) repère ce qui lui semble être une erreur d'énoncé, il (elle) le signale très lisiblement sur sa copie, propose la correction et poursuit l'épreuve en conséquence.

De même, si cela vous conduit à formuler une ou plusieurs hypothèses, il vous est demandé de la (ou les) mentionner explicitement.

NB : ***La copie que vous rendrez ne devra, conformément au principe d'anonymat, comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé comporte notamment la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de signer ou de l'identifier.***

A

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

Constitution du sujet

Les documents réponses DR1 à DR6 (pages 28 à 32) doivent être rendus avec la copie.

Contexte de l'étude

Le karting électrique est une activité sportive automobile respectueuse de l'environnement (pas de consommation de produits pétroliers, sans odeur), et totalement silencieuse.
Les véhicules utilisés pour l'activité de karting sont nommés karts et offrent des sensations de conduite uniques de vitesse, d'accélération et de glisse.

La société support de cette étude possède une piste de location de karts électriques homologuée par la FFSA (Fédération Française du Sport Automobile).

Afin de s'adapter aux différents niveaux des pratiquants, la société a opté pour une régulation des vitesses maximales des karts ajustable à distance. Plusieurs vitesses de course peuvent être configurées sur les karts : V1 (enfants de moins de 14 ans), V2 (jeunes de plus de 14 ans et adultes), V3 (pilotes ayant réalisé un tour de circuit en moins de 21 secondes en V2). Tous ces points permettent de mettre en confiance un large public et de rendre accessible cette activité au plus grand nombre.

Pour mesurer l'évolution du niveau des pratiquants et permettre l'organisation de compétitions, un système de chronométrage automatique a été installé. Celui-ci fonctionne avec des systèmes de détections de passage sur la ligne d'arrivée, dont deux versions existent :

- la première utilise une boucle métallique noyée dans la dalle de la piste, permettant à des transpondeurs placés sur les karts de mesurer les temps réalisés ;
- la seconde utilise des tags RFID installés sur les karts, détectés par une antenne placée sur un portique au niveau de la ligne d'arrivée.

Un système informatique permet au responsable du karting de gérer les temps réalisés par les pilotes, les niveaux de batterie des karts et d'autres éléments détaillés par la suite.

L'objectif de cette étude est, dans un premier temps, de justifier la première architecture de chronométrage utilisée par la société de karting, puis de concevoir une partie du système de gestion de course et de mise en réseau des différents éléments, afin d'assurer les prestations énoncées ci-avant. Enfin, une comparaison avec la seconde architecture de chronométrage sera réalisée avant d'envisager des évolutions possibles du fonctionnement de l'activité.

Partie 1 – Analyse préliminaire du système

Objectif : analyser les différents besoins du système et définir l'architecture du karting.

Les utilisateurs achètent des séries ou sessions correspondant au déroulement d'une course entre plusieurs participants durant une durée déterminée.

Le karting dispose de 20 karts. Durant le déroulement de la série, 8 karts au maximum peuvent rouler simultanément. Une série peut comporter au minimum un kart, dans le cas d'un entraînement. Les karts non utilisés sont en « rechargement batterie ».

Chaque série est chronométrée. Le meilleur temps réalisé est enregistré et entre instantanément dans le classement.
Les résultats sont consultables sur place via des écrans situés sur la piste et à l'accueil. Les participants inscrits comme membre du karting peuvent éventuellement consulter les statistiques (temps, classements, vainqueurs) sur le site web du karting à l'aide de leur mot de passe. Les scores sont établis par série, mois, et année.

Le commissaire de course est un membre du personnel du karting en charge de la sécurité des clients participant aux courses, et de son bon déroulement. Les participants s'enregistrent à l'accueil pour une série donnée et règlent leur(s) série(s) auprès du commissaire de course s'il est disponible ou auprès d'un employé du karting.

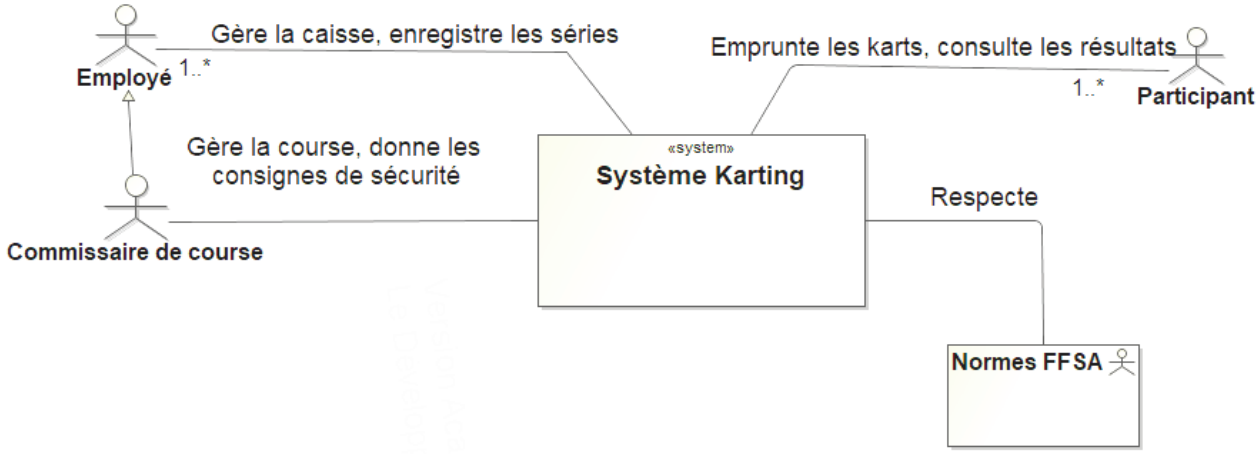


Diagramme de contexte du karting

Question 1. Préciser et **justifier** le type de relation UML existant entre l'acteur « Commissaire de course » et l'acteur « Employé » sur le diagramme de contexte du karting.

Le système informatique du karting est composé :

- d'une piste avec une boucle métallique qui permet de détecter le passage des karts à chaque tour ;
- de 20 karts électriques munis chacun d'un transpondeur sur la face avant. Les transpondeurs sont configurés pour avoir le même numéro que les numéros de karts ;
  d'un boîtier décodeur qui traite les informations détectées au passage d'un transpondeur et les transmet au « PC Piste » via le port USB ;
- d'un ordinateur « PC Piste » situé au bord de piste au niveau de la ligne de départ. Il pilote le décodeur, transmission des ordres de start et stop, acquisition des trames lors du passage des karts (Document technique DT1) ;
- de deux écrans reliés au « PC Piste », un situé sur la piste et un situé à l'accueil, qui permettent de suivre les courses en temps réel. Ils affichent les temps effectués par les concurrents en temps réel pour chaque tour, ainsi qu'un classement du meilleur temps de chaque client pour la série en cours ;

Tournez la page S.V.P.

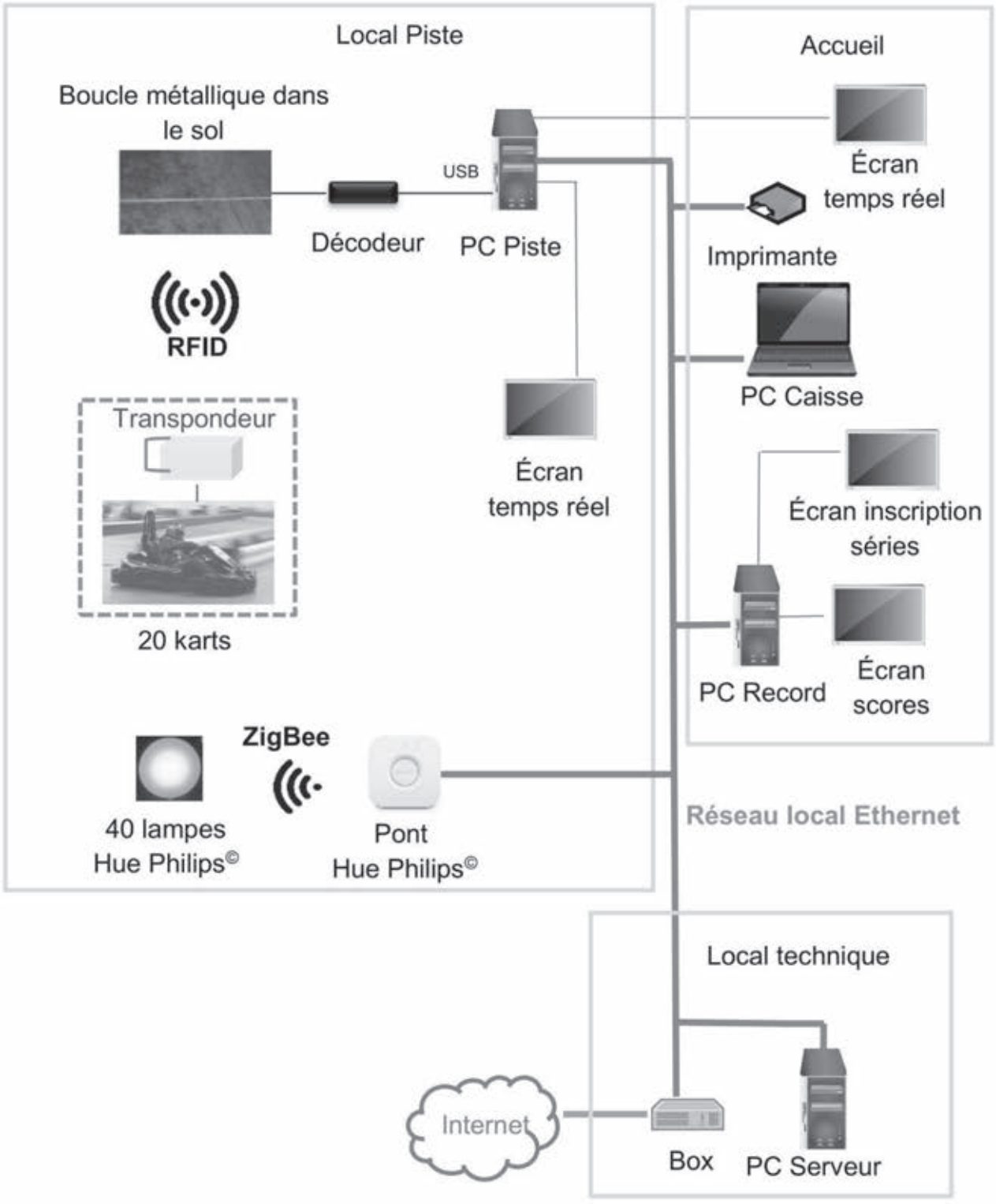
- d’un système d’éclairage connecté composé de 40 lampes basse consommation à LED hues (voir document technique DT8) dont la couleur varie en fonction du temps lors de la course et à chaque fois qu’un passage de kart est détecté ;
- d’un ordinateur « PC Serveur » situé dans un local technique qui est accessible depuis Internet pour la lecture des résultats. Il dispose d’un serveur web Apache et d’une base de données Mysql contenant les résultats de toutes les courses ;
- d’un ordinateur « PC Record » situé à l’accueil ;
- de deux écrans reliés au « PC Record » situés à l’accueil qui permettent respectivement d’afficher sur un écran les scores et sur un autre écran les séries et les pseudos des personnes inscrites ;
- d’un « PC Caisse » situé à l’accueil permettant de gérer les paiements des séries et des activités du complexe et d’enregistrer les concurrents pour une série. Une imprimante en réseau permet d’imprimer des tickets de paiement ou des résultats de course ;
- d’équipements réseau constitués d’un commutateur administrable et d’une box Internet.

Ainsi, le commissaire de course peut visualiser les séries à réaliser depuis « le PC Piste ». Il appelle les participants pour une série donnée à l’aide d’un haut-parleur et gère la course. Il donne les ordres de départ et de fin de course aux participants.

Durant la course, le décodeur transmet à chaque passage des karts sur la boucle métallique les temps de passage et les numéros de transpondeurs au « PC Piste ». Les pseudos des participants et les temps de chaque tour sont affichés durant la course sur les écrans du « PC Piste » et enregistrés dans la base de données du « PC Serveur ».

À partir des données enregistrées par le « PC Piste » dans la base de données du « PC Serveur », le « PC Record » est chargé d’afficher les statistiques sur des écrans à partir d’un navigateur. Le traitement est réalisé sur des pages Web codées en Php, implémentées sur le « PC serveur ».

Le synoptique ci-après décrit le système informatique du karting :



Question 2. Sur le document réponse DR1, **compléter** le diagramme de cas d'utilisation. **Indiquer** les cardinalités des relations si elles sont différentes de 1.

Question 3. Sur le document réponse DR2, **compléter** le diagramme de déploiement. **Indiquer** les noms des supports transportant l'information (Ethernet, Zigbee, RFID, USB) et les cardinalités des liaisons si elles sont différentes de 1.

Tournez la page S.V.P.

Question 4. Sur le document réponse DR3, **compléter** le diagramme de séquence de manière à montrer l'affichage en temps réel des temps effectués sur chaque tour et l'enregistrement des mesures en bases de données.

Partie 2 – Transmission des données entre le boîtier « Décodeur » et le « PC Piste »

Objectif : étudier le protocole du boîtier décodeur par rapport aux besoins de mesurage.

Le boîtier décodeur sert à traiter les informations détectées par la boucle métallique au passage d'un transpondeur et les transmet au « PC Piste » via le port USB afin de réaliser la mesure des temps au tour réalisés par les pilotes.

Question 5. Après avoir consulté la documentation relative au décodeur (document technique DT1), **établir** la trame qui doit être transmise au décodeur au départ de la course et **calculer** la durée de transmission de cette trame.

Les temps de passage des karts sont établis à partir de l'instant de la réception de la trame Start par le décodeur. Les temps de passage représentent donc la durée écoulée entre la réception de l'ordre de Start par le décodeur et le passage du kart sur la boucle métallique. Les karts démarrent en amont de la boucle métallique.
Le temps de passage mesuré pour le premier passage sur la boucle métallique du kart 1 est t_0 = 12 453 ms. Il s'agit du premier passage sur la boucle métallique mais aucun tour n'a été effectué.
Le temps de passage mesuré pour le deuxième passage sur la boucle métallique du kart 1 est t_1 = 37 542 ms.

Question 6. Calculer la durée du tour numéro 1. La durée de transmission de la trame calculée à la question 5 **provoque-t-elle** une erreur sur les valeurs des temps au tour calculées ?

Question 7. Après avoir consulté la documentation relative au décodeur (document technique DT1) et la table des codes ASCII (document technique DT2), **établir** la trame émise par le décodeur lors du passage d'un kart avec les informations suivantes :

- boucle (loop), numéro 1 ;
- kart, numéro 2 ;
- niveau d'émission de la boucle magnétique (heat), 69 % ;
- niveau de batterie, 3 ;
- temps écoulé entre la réception de l'ordre de départ et le passage du kart sur la boucle métallique, 10 secondes.

Question 8. Justifier la présence du champ CCC dans la trame.

Question 9. Le protocole du décodeur permet-il d'**obtenir** les informations nécessaires pour répondre aux besoins de mesurage des temps au tour réalisés par chaque kart ? **Justifier** numériquement l'étendue des mesures possibles.

Partie 3 – Conception détaillée : gestion d'une course depuis le « PC Piste »

Objectif : gérer le départ et la fin d'une course et extraire les données réceptionnées lors du passage des karts sur la boucle métallique. Enregistrer les mesures en base de données.

L'application du « PC Piste » permet au commissaire de course, à partir d'une IHM, d'effectuer les opérations suivantes :

- connexion au décodeur ;
- saisie du numéro de série ;
- démarrage de la course et acquisition des temps de chaque tour pour tous les karts ;
- arrêt de la course.

Les mesures des temps au tour sont ainsi enregistrées en base de données MySQL sur le « PC Serveur », qui pourra ensuite servir à restituer les temps réalisés aux clients désireux de mesurer leurs performances.

L'application du « PC Piste » est réalisée en programmation orientée objet en langage C#. Le diagramme de classes (document technique DT3) décrit les classes utilisées. Le document technique DT4 rappelle quelques instructions courantes utilisées dans ce langage.

Les classes SerialPort, Queue, String et Convert, sont des classes de la bibliothèque .NET Framework. Elles sont décrites dans le document technique DT5.

Pour rappel, le document technique DT1 contient la documentation relative au décodeur. La classe Chronomètre est chargée de dialoguer avec le décodeur grâce à la classe SerialPort qui contient toutes les fonctions de gestion d'un port série.

Question 10. La classe SerialPort est une classe de stéréotype « Boundary ». **Indiquer** ce que cela signifie.

Question 11. Indiquer le type de relation UML existant entre la classe CChronometre et la classe SerialPort.

Question 12. Indiquer le type de relation UML existant entre la classe Form1 et la classe Form.

Question 13. Écrire en C# la déclaration de la classe CChronometre en laissant le corps des méthodes vides.

Question 14. La classe Commun contient des attributs publics statiques qui pourront être utilisés par différentes classes de l'application. **Préciser** la définition d'un attribut statique et ce que cela implique dans son utilisation.

La classe Commun a été implémentée de la façon suivante :

```
public class Commun
{
    public static Queue<SMesure> fileMesure = new Queue<SMesure>();
    public static int nEtat = (int)ETAT.DEBUT;
}
```

Question 15. Préciser le rôle de l'opérateur new.

Question 16. La classe Queue est une file d'attente. **Décrire** le principe de fonctionnement d'une file d'attente. **Citer** deux autres types de collection.

Question 17. La classe Queue est une classe générique. **Indiquer** la signification de « classe générique ».

Question 18. Citer une autre bibliothèque qui implémente des collections génériques en C++ et qui fonctionne sous Windows et sous Linux.

Question 19. Donner en C# l'implémentation de la méthode OrdreStart() qui transmet l'ordre Start au décodeur. Cette méthode intercepte les exceptions déclenchées par SerialPort et génère une exception en cas d'erreur avec le message « Erreur écriture ligne série ».

Question 20. Indiquer les trames du décodeur qui peuvent être retournées par la méthode ReadLine() de la classe SerialPort dans les cas suivants :

- départ de la course ;
- fin de la course ;
- passage d'un kart sur la boucle métallique (en reprenant le cas étudié à la question 7).

La méthode CheksumOk() de la classe CChronometre permet de tester le checksum de la trame « sTrame » passée en paramètre. Cette trame a été obtenue par la méthode ReadLine() de SerialPort lors du passage du kart. Cette méthode retourne la valeur « true » si le checksum de la trame est correct, « false » sinon.

Question 21. Donner en C# l'implémentation de la méthode CheksumOk() de la classe CChronometre.

La méthode serialPort1_DataReceived() de la classe CChronometre est déclenchée sur l'évènement réception d'un caractère sur le port USB du « PC Piste ».

Question 22. Donner la définition d'une programmation évènementielle et citer d'autres types d'évènements.

ETAT est une énumération permettant de mémoriser l'état de la course. Dans l'état DEBUT la course n'est pas démarrée. Dans l'état MESURE la course est en cours.
La déclaration de ETAT est la suivante :

```
enum ETAT
{
        DEBUT = 1,
        MESURE = 2
}
```

L'algorithme de la méthode serialPort1_DataReceived() est précisé dans le diagramme de séquence du document technique DT6.

Question 23. À partir du diagramme de séquence présenté dans le document technique DT6, **compléter** en C# l'implémentation de la méthode serialPort1_DataReceived() sur le document réponse DR4.

Le code source C# est compilé par le compilateur en un code intermédiaire qui sera ensuite exécuté par une machine virtuelle qui dépend du système d'exploitation.

Question 24. Indiquer si une machine virtuelle est nécessaire pour l'exécution d'un programme dont la compilation et l'édition de liens ont été effectués à partir d'un code source écrit en C++. **Justifier**.

Les renseignements concernant les clients et les mesures des temps de passage sont enregistrés en base de données. Le document technique DT7 donne une description partielle de la base de données ainsi qu'un extrait des données enregistrées.

Question 25. Compléter le document réponse DR5 en ajoutant la ligne dans la table TTourClient dans le cas suivant :

- le client a pour pseudo « Polo » ;
- le client a participé à la série numéro 40 ;
- le client a réalisé le premier passage sur la boucle métallique après son départ en amont de la boucle, le tour est numéroté 0 ;
- la trame émise par le décodeur est celle qui est détaillée à la question 7.

Afin d'enregistrer les mesures correspondant à un passage de kart sur la boucle métallique, il faut rechercher le numéro de série idSérieClient pour un numéro de kart et un numéro de série donné.

Question 26. Écrire en langage SQL la requête permettant de rechercher le numéro de série idSérieClient pour le kart numéro 2 et le numéro de série 40.

Question 27. Écrire en langage SQL la requête permettant d'insérer la ligne définie à la question 25.

L'interface IHM du « PC Piste » est une boîte de dialogue implémentée dans la classe Form1 décrite dans le diagramme de classes du document technique DT3. La méthode RunCourse() est une méthode lancée lorsque le thread « threadCourse » est démarré.

Pour rappel, les mesures sont enregistrées dans une file d'attente dans la méthode serialPort1_DataReceived() de la classe CChronometre. Le traitement de la méthode RunCourse() consiste à lire les données enregistrées dans la file d'attente (opération Dequeue() de la classe Queue) et à les enregistrer dans la base de données grâce aux méthodes de la classe CBddChrono. La lecture de la file d'attente peut donc être interrompue par son écriture ou vice-versa.

Question 28. Proposer une solution et un algorithme pour résoudre ce problème d'écriture/lecture dans la file d'attente.

Question 29. Conclure en justifiant le traitement de la réception sur ligne série par évènement et le traitement de la méthode RunCourse dans un thread.

Partie 4 – Mise en réseau des équipements

Objectif : étudier la mise en réseau des équipements permettant d'accéder aux données sur les différents lieux du karting et depuis l'extérieur.

Le réseau du karting est à l'adresse IP 192.168.1.0/24. Le serveur DHCP est activé sur la box.
Le plan d'adressage des différentes machines du réseau permettant de gérer l'activité doit suivre les règles suivantes :

- la première adresse hôte valide du réseau est attribuée au routeur, ici il s'agit de l'adresse LAN de la box ;
- la deuxième adresse hôte valide est attribuée au commutateur administrable ;
- la troisième adresse hôte valide du réseau est attribuée à l'imprimante ;
- la seizième adresse hôte valide est attribuée au serveur ;
- la dix-septième adresse hôte valide est attribuée au pont Phillips© Hue.

Les autres équipements auront leur adresse réseau attribuée par le serveur DHCP. La plage DHCP est fixée de 192.168.1.20 à 192.168.1.150.

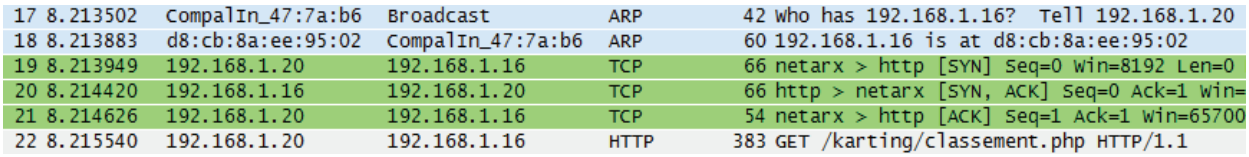
Question 30. Indiquer ce que signifie /24 pour une adresse réseau 192.168.1.0/24.

Question 31. Calculer le nombre d'hôtes possible pour ce réseau.

Question 32. Établir le plan d'adressage pour tous les hôtes du réseau. **Justifier** le choix concernant l'adressage statique ou dynamique des hôtes.

Pour afficher les résultats depuis le « PC Record », l'URL http://192.168.1.16/karting/classement.php a été saisie dans le navigateur.

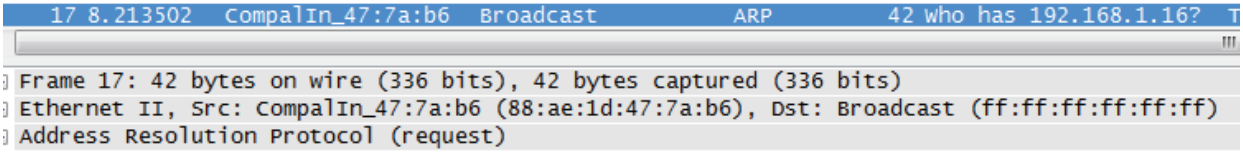
Un relevé des trames sous Wireshark a été effectué lors de l'échange.



17	8.213502	CompalIn_47:7a:b6	Broadcast	ARP	42 Who has 192.168.1.16? Tell 192.168.1.20
18	8.213883	d8:cb:8a:ee:95:02	CompalIn_47:7a:b6	ARP	60 192.168.1.16 is at d8:cb:8a:ee:95:02
19	8.213949	192.168.1.20	192.168.1.16	TCP	66 netarx > http [SYN] Seq=0 Win=8192 Len=0
20	8.214420	192.168.1.16	192.168.1.20	TCP	66 http > netarx [SYN, ACK] Seq=0 Ack=1 Win=
21	8.214626	192.168.1.20	192.168.1.16	TCP	54 netarx > http [ACK] Seq=1 Ack=1 Win=65700
22	8.215540	192.168.1.20	192.168.1.16	HTTP	383 GET /karting/classement.php HTTP/1.1

Question 33. Indiquer le rôle des trames 17 et 18.

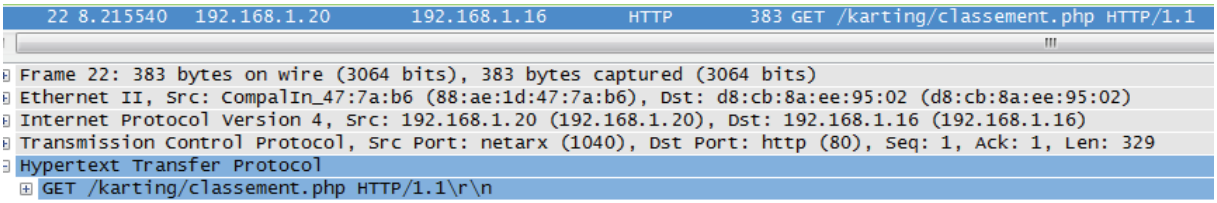
Un relevé détaillé de la trame 17 est donné ci-dessous.



```
17 8.213502  CompalIn_47:7a:b6  Broadcast          ARP        42 Who has 192.168.1.16?  T
Frame 17: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
Ethernet II, Src: CompalIn_47:7a:b6 (88:ae:1d:47:7a:b6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
```

Question 34 Justifier la valeur de ff :ff :ff :ff :ff pour l'adresse mac de destination. **Indiquer** quels équipements réseaux transmettent une requête ARP.

Un relevé détaillé de la trame 22 est donné ci-dessous.



```
22 8.215540  192.168.1.20      192.168.1.16       HTTP       383 GET /karting/classement.php HTTP/1.1
Frame 22: 383 bytes on wire (3064 bits), 383 bytes captured (3064 bits)
Ethernet II, Src: CompalIn_47:7a:b6 (88:ae:1d:47:7a:b6), Dst: d8:cb:8a:ee:95:02 (d8:cb:8a:ee:95:02)
Internet Protocol Version 4, Src: 192.168.1.20 (192.168.1.20), Dst: 192.168.1.16 (192.168.1.16)
Transmission Control Protocol, Src Port: netarx (1040), Dst Port: http (80), Seq: 1, Ack: 1, Len: 329
Hypertext Transfer Protocol
  GET /karting/classement.php HTTP/1.1\r\n
```

Question 35. Compléter le document réponse DR6 en indiquant pour chaque couche du modèle OSI sur lesquelles repose la trame 22 :

- le nom de la couche dans le modèle OSI ;
- le protocole utilisé ;
- les principales informations transmises dans cette couche.

Les participants inscrits comme membres du karting peuvent, munis de leur mot de passe, consulter les statistiques (temps, classements, vainqueurs) sur le site web du karting. Le « PC Serveur » est configuré dans une DMZ grâce à une box « pro ».

Question 36. Indiquer ce qu'est une DMZ.

La mise en lumière de la piste du karting est réalisée grâce à des lampes à LED Philips© Hue (document technique DT8). Le pilotage des lampes est réalisé depuis le PC Piste au moyen de services REST implémentés sur le pont Philips© Hue. Le document technique DT8 donne les caractéristiques des lampes et une implémentation de leur pilotage en langage C#.

Question 37. Indiquer les caractéristiques du service REST implémenté.

Le test du réseau par des commandes ping a donné pour tous les équipements des réponses de ce type :

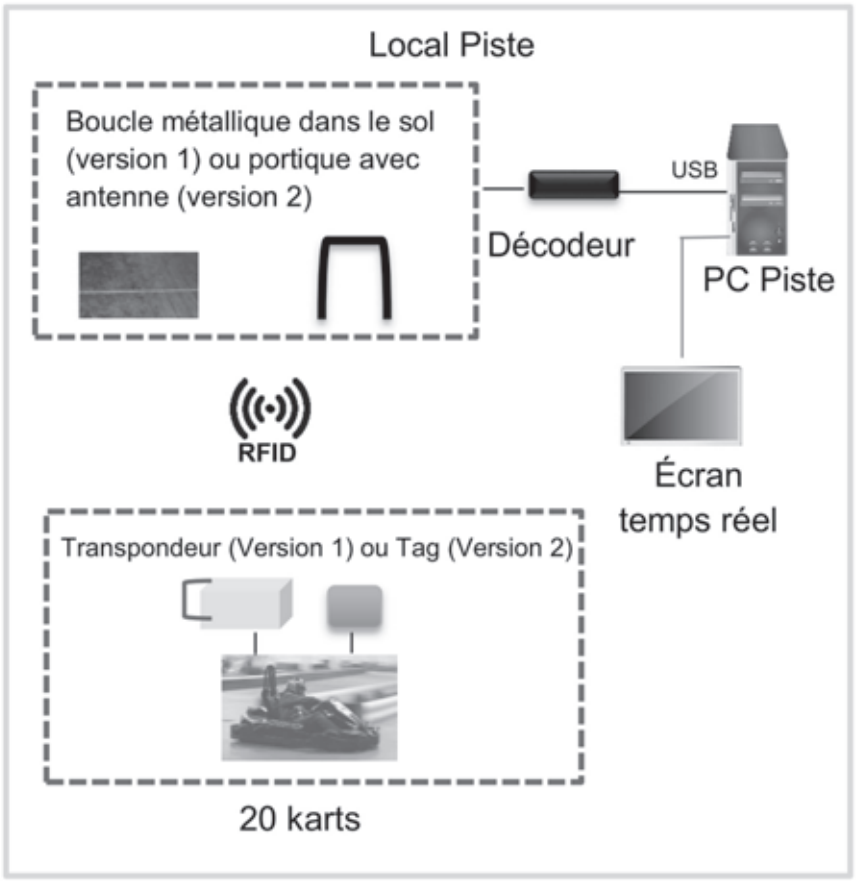
```
C:\>ping 192.168.1.1

Envoi d'une requête 'Ping'  192.168.1.1 avec 32 octets de données :
Réponse de 192.168.1.1 : octets=32 temps=1 ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64
Réponse de 192.168.1.1 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 192.168.1.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms
```

Question 38. Conclure sur la capacité du réseau à assurer la communication des équipements nécessaires aux besoins du karting.

Partie 5 – Synthèse

Objectif : faire le bilan de la solution étudiée et proposer des évolutions.

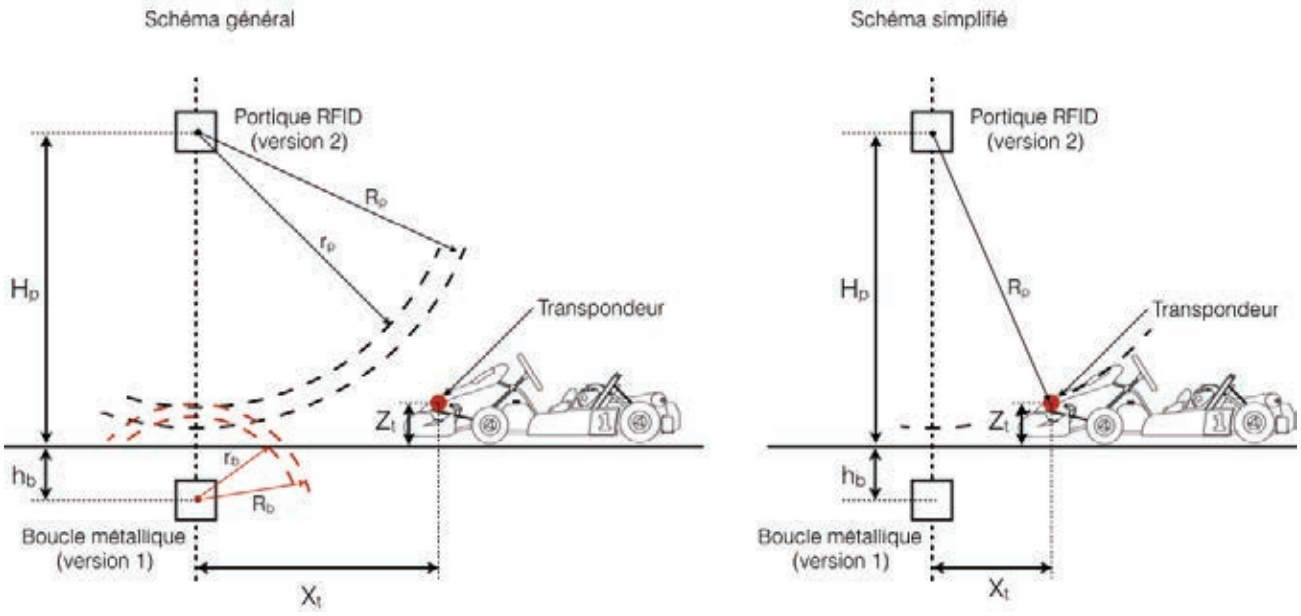
Une évolution envisagée est d'équiper les karts de tags RFID et d'installer un portique muni d'une antenne sur la ligne d'arrivée. Un nouveau décodeur permet alors de transmettre à un PC nommé « PC Piste » les numéros des tags RFID et la date/heure lors du passage des karts. Contrairement à la version précédente, il n'y a pas d'ordre « départ course » et « fin course ». Le décodeur est remis à l'heure depuis le « PC Piste » auquel il transmet la date et l'heure réelle.



Question 39. Il est prévu pour les systèmes de détection des karts d'évoluer vers cette nouvelle version. **Comparer** les deux solutions au niveau de l'installation du système et du protocole.

Pour départager les deux versions de chronométrage, on souhaite vérifier la précision de chacune d'entre elles. Le karting souhaite transmettre aux pilotes des chronométrages précis au centième de seconde. Pour la suite, on considère les schémas suivants, dans lesquels sont représentés :

- la ligne de départ/arrivée en pointillés verticaux ;
- les rayons r_p et R_p correspondant aux rayons mini et maxi de détection via le portique RFID situé à une hauteur H_p ;
- les rayons r_b et R_b correspondant aux rayons mini et maxi de détection via les boucles métalliques noyées dans la piste (profondeur d'enfouissement h_b) ;
- le transpondeur monté sur le kart situé à une distance X_t de la ligne de départ/arrivée et à une hauteur Z_t du sol.



Question 40. Exprimer, en fonction des paramètres précédents, la variation de distance ΔX_{tp} correspondant à la différence entre la position du kart lors la détection par le portique RFID dans le cas où la détection se fait à un rayon R_p, puis dans le cas où la détection se fait à un rayon r_p.

Question 41. Faire de même pour la détection via la boucle métallique en exprimant la variation de distance ΔX_{tb} correspondant à la différence entre la position du kart lors la détection par la boucle métallique dans le cas où la détection se fait à un rayon R_b, puis dans le cas où la détection se fait à un rayon r_b.

Les applications numériques seront réalisées avec les valeurs suivantes :
R_p = 3 m ; r_p = 2,95 m ; R_b = 1 m ; r_b = 0,98 m ; H_p = 2 m ; Z_T = 0,5 m ; h_b = 3 cm.

Tournez la page S.V.P.

Question 42. Exprimer alors l'erreur temporelle ΔT réalisée dans les deux cas. **Effectuer** l'application numérique en supposant que le kart passe à une vitesse de 30 $km \cdot h^{-1}$. **Conclure** vis-à-vis du critère de précision des deux solutions.

Le responsable du karting souhaite faire évoluer l'acquisition du niveau de charge des karts en implémentant sur chaque véhicule une carte de mesure du niveau de la batterie compatible Sigfox© et alimentée par une pile. Le document technique DT9 fournit les principales caractéristiques du réseau Sigfox©.

Question 43. Décrire la manière dont est réalisée la mesure du niveau de batterie des karts dans le système actuel. **Indiquer** l'avantage apporté par le nouveau système.

Question 44. Conclure en indiquant les éléments à mettre en place pour une gestion distante des résultats des courses et des niveaux de batterie des karts.

DOCUMENTS TECHNIQUES

Document technique DT1

Partie 1, 2 et 3 – Protocole du décodeur version 1

(Document sur 2 pages)

Le décodeur est relié au port USB du PC via un adaptateur USB et est vu coté PC comme un port COM RS232.

Caractéristiques du port série : 9 600 bauds, 8 bits de donnés, 1 bit de stop, sans parité, pas de contrôle de flux.

Toutes les trames sont codées en caractères ascii.

Ordre de start :

Données transmises du PC vers le décodeur :

```
$ST
```

Données transmises du décodeur vers le PC :

```
$M;START\r\n
```

Ordre de stop :

Données transmises du PC vers le décodeur :

```
$AR
```

Données transmises du décodeur vers le PC :

```
$M;STOP\r\n
```

Passage d'un kart sur la boucle métallique :

Données transmises du décodeur vers le PC :

```
$P;BB;NNNNN;HH;V;TTTTTTTTT;CCC\r\n
```

$P : caractère '$' suivi du caractère 'P'
BB : numéro de la boucle.
NNNNN : numéro du transpondeur.
HH : Niveau d'émission de la boucle métallique (heat).
V : niveau batterie
TTTTTTTTT : Temps en ms écoulés depuis la réception de l'ordre de start.
CCC : Cheksum.

Les caractères B, N, H, T, C sont compris entre '0' et '9'.
Le caractère V est compris entre '1' et '4'.
'x' signifie « code ascii de x », par exemple '0' signifie « code ascii de 0 ».
\r est le caractère « Carriage Return », codé 13 en décimal en Ascii.
\n est le caractère « Line Feed », code 10 en décimal en Ascii.

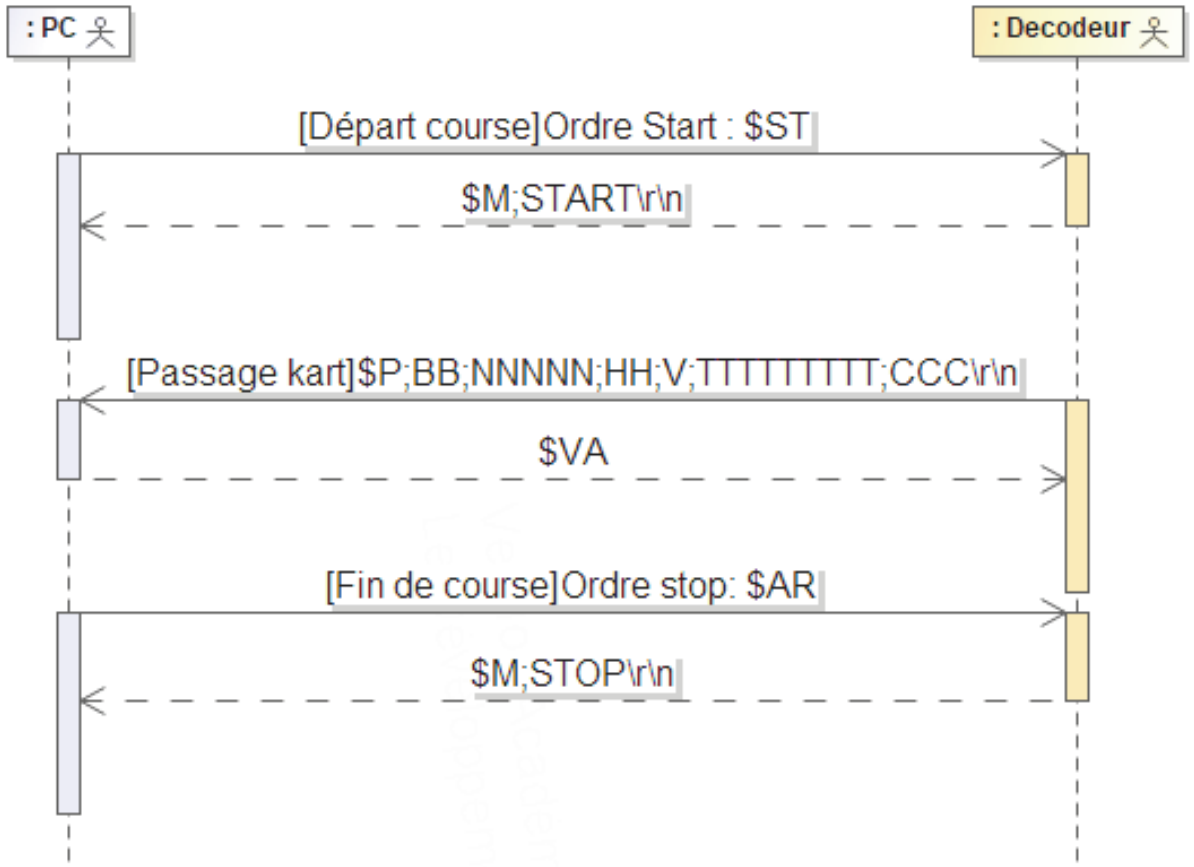
Calcul du cheksum CCC : ou exclusif des 27 premiers octets codés ascii de la trame suivi d'un complément à 1. Un complément à 1 consiste à inverser tous les

bits. Le nombre décimal obtenu est ensuite converti en une chaîne ascii notée CCC sur 3 caractères.

Données transmises du PC vers le décodeur :

`$VA`

Acquittement de la trame.



Document technique DT2

Partie 2 **–** Table des codes ASCII

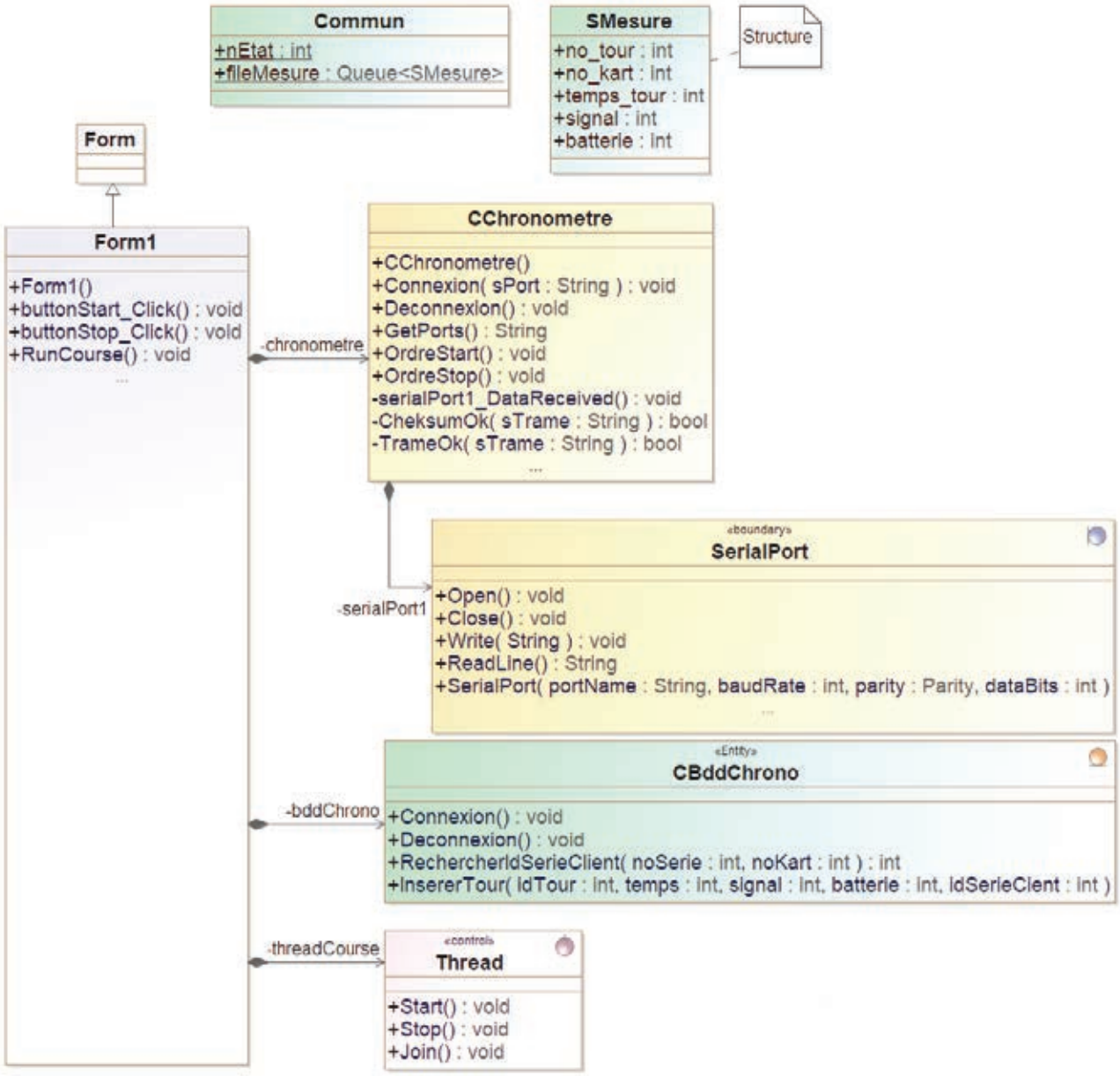
ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
0	0	NUL	16	10	DLE	32	20	(space)	48	30	0
1	1	SOH	17	11	DC1	33	21	!	49	31	1
2	2	STX	18	12	DC2	34	22	"	50	32	2
3	3	ETX	19	13	DC3	35	23	#	51	33	3
4	4	EOT	20	14	DC4	36	24	$	52	34	4
5	5	ENQ	21	15	NAK	37	25	%	53	35	5
6	6	ACK	22	16	SYN	38	26	&	54	36	6
7	7	BEL	23	17	ETB	39	27	'	55	37	7
8	8	BS	24	18	CAN	40	28	(	56	38	8
9	9	TAB	25	19	EM	41	29	)	57	39	9
10	A	LF	26	1A	SUB	42	2A	*	58	3A	:
11	B	VT	27	1B	ESC	43	2B	+	59	3B	;
12	C	FF	28	1C	FS	44	2C	,	60	3C	<
13	D	CR	29	1D	GS	45	2D	-	61	3D	=
14	E	SO	30	1E	RS	46	2E	.	62	3E	>
15	F	SI	31	1F	US	47	2F	/	63	3F	?

ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol	ASCII	Hex	Symbol
64	40	@	80	50	P	96	60	`	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[	107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	|
77	4D	M	93	5D	]	109	6D	m	125	7D	}
78	4E	N	94	5E	^	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	

Le caractère LF (Line Feed) est noté en C# ‘\n’
Le caractère CR (Carriage Return) est noté en C# ‘\r’

Document technique DT3

Partie 3 – Diagramme de classes de l'application PC Piste



Note concernant le codage en C# :

- les variables de type SMesure sont créées sur la pile ;
- les objets des classes sont créés sur le tas managé (avec new).

```
// Exemple en mode console
 class CLigneSerie
 {
     SerialPort portSerie;
     // Constructeur paramétré
     // sPort : nom du port série COMx
     public CLigneSerie(String sPort)
     {
       // crée le port série 9600 bds 8 bits données 1 stop bit
       // pas de parité
       this.portSerie = new SerialPort(sPort, 9600,
       Parity.None, 8, StopBits.One);
     }
     public void Connexion()// se connecte au port série
     {
        try
        {
          this.portSerie.Open(); // ouvre le port
        }
        catch (Exception exe)
        {
          // génère une exception si erreur
          throw new Exception("Erreur connexion ligne série\r\n");
        }
     }
  }
  class Program
  {
      static void Main(string[] args)
      {
        string sNomKart = "Kart no 15";
        string sNumero = sNomKart.Substring(8, 2); // extrait 15
        Console.WriteLine(sNumero);                // Affiche 15
        int nNumero = Convert.ToInt32(sNumero);    // en entier
        Console.WriteLine(nNumero);
        byte byNumero = Convert.ToByte(sNumero);      // en octet
        Console.WriteLine(byNumero);
        byte byOctet = (byte) sNomKart[0];     //byteOcet vaut 'K'
        CLigneSerie ligneSerie = new CLigneSerie("COM1");
        try
        {
            ligneSerie.Connexion();
        }
        catch (Exception exe)     // Intercepte l'exception
        {   // Affiche "Erreur connexion ligne série \r\n"
            Console.WriteLine(exe.Message);
        }
      }
   }
```

Document technique DT5

Partie 3 – Extraits de la documentation de la bibliothèque .NET Framework

(Document sur 2 pages)

Classe SerialPort

Constructeurs :

`SerialPort();`

Initialise une nouvelle instance de la classe SerialPort.

`SerialPort(String, Int32, Parity, Int32, StopBits);`

Initialise une nouvelle instance de la classe SerialPort avec le nom de port, la vitesse (en bauds), le bit de parité, les bits de données et le bit d'arrêt spécifiés.

Méthodes :

`Open();`

Ouvre une nouvelle connexion de port série.

`Close();`

Ferme la connexion du port.

`Write(String);`

Écrit la chaîne spécifiée sur le port série.

`String ReadLine();`

Lit jusqu'à la valeur NewLine dans la mémoire tampon d'entrée.

Par défaut, la valeur NewLine vaut '\n'. La chaîne retournée contient toutes les valeurs lues avant le caractère '\n' donc le caractère '\n' ne fait pas partie de la chaîne retournée.

Évènements

`DataReceived`

Indique que des données ont été reçues via un port représenté par l'objet SerialPort.

Exceptions

`Exception`

Permet d'intercepter tout type d'exception. Les méthodes décrites ci-dessus génèrent toutes des exceptions lorsque l'opération s'est mal terminée.

Classe Queue<T>

Représente une collection d'objets type « file d'attente »

Constructeur

`Queue<T>() ;`

Initialise une nouvelle instance de la classe Queue<T> qui est vide

Méthodes

`T Dequeue();`

Supprime et renvoie l'objet au début de la Queue<T>.

`Enqueue(T) ;`

Ajoute un objet à la fin de la Queue<T>.

Classe String

Constructeurs

`String(Char*);`

Initialise une nouvelle instance de la classe String à la valeur indiquée par un pointeur spécifié vers un tableau de caractères Unicode.

Propriétés

`Length`

Obtient le nombre de caractères de l'objet String actuel.

Méthodes

`String Substring(Int32 pos, Int32 longueur);`

Retourne une sous-chaîne de l'objet String actuel. La sous-chaîne commence à la position spécifiée par le premier paramètre « `pos` » et à pour longueur le paramètre « `longueur` ».

Opérateurs

+ (concaténation), = (affectation), == != (tests), [ ] (élément de la chaîne, en lecture seulement)

Classe Convert

Méthodes

`static byte ToByte(String s) ;`

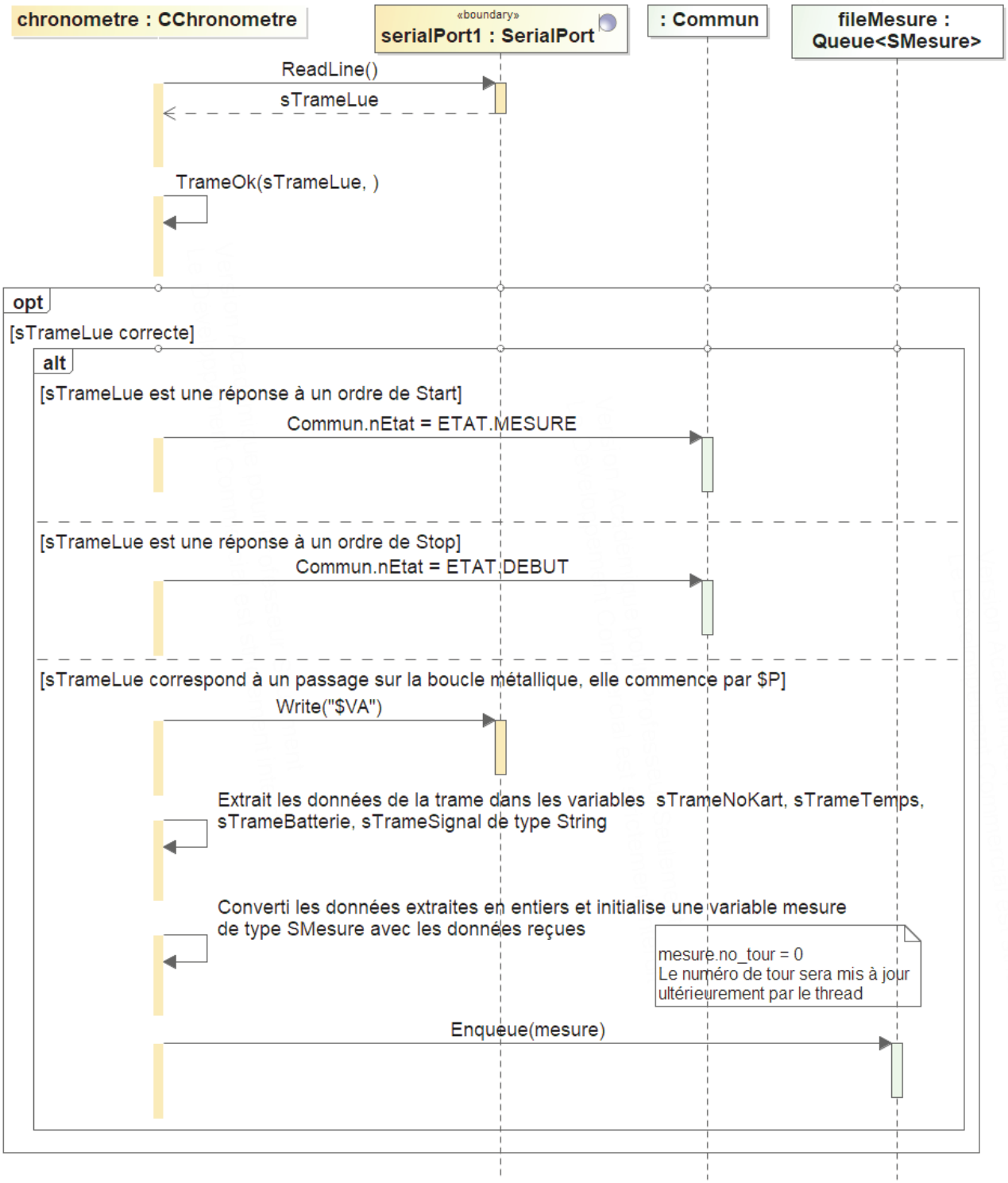
Convertit la String « `s` » représentant un nombre en base 10 en un entier non signé sur 8 bits. Le type byte est équivalent au type « unsigned char » du C++.

`static int ToInt32(String s);`

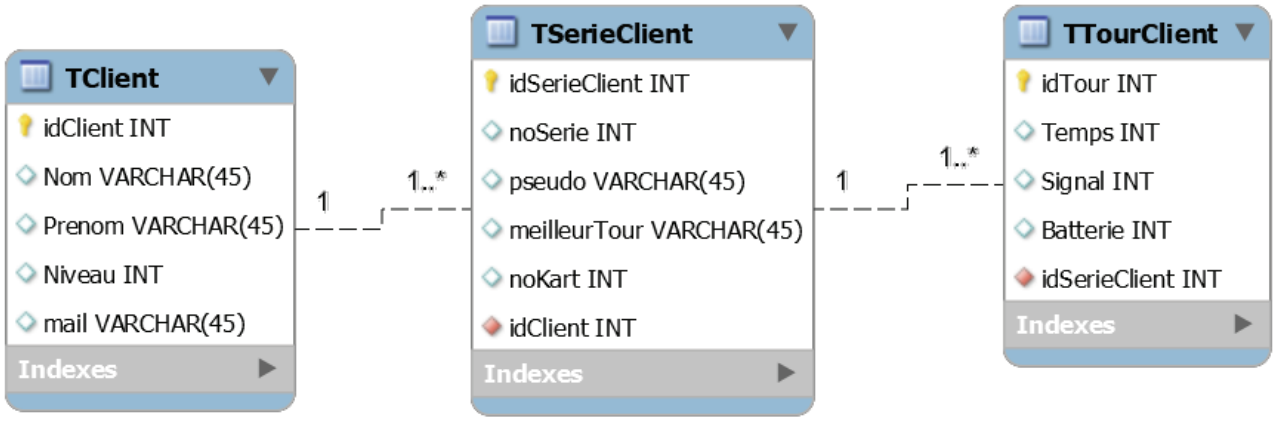
Convertit la String s représentant un nombre en base 10 en un entier sur 32 bits.

Document technique DT6

Partie 3 – Diagramme de séquence de la méthode serialPort1_DataReceived()



Document technique DT7
Partie 3 – Modélisation de la base de données



Le type de modèle est ici UML.

Le symbole représenté par une clé indique que le champ est une clé primaire : TClient.idClient, TSerieClient.idSerieClient, TTourClient.idTour sont des clés primaires.
Le symbole représenté par un losange plein indique que le champ est une clé secondaire : TSerieClient.idClient, TTourClient.idSerieClient sont des clés secondaires.

La table TClient contient les données concernant les clients.

La table TSerieClient contient les données concernant les séries effectuées par un client. Le client peut changer de pseudo à chaque série qu'il effectue. Le champ « IdSerieClient » est un numéro unique pour un client qui a effectué une série. Le champ « noSerie » est le numéro de la série. Tous les clients qui auront effectué la même série auront le champ « noSerie » identique. Le champ idSerieClient est unique pour un numéro de série « noSerie » et un pseudo donné.

La table TTourClient contient les résultats de chaque tour pour une série donnée et un client.
Le champ « Signal » est le niveau d'émission de la boucle magnétique, le champ « Batterie» est le niveau de batterie, le champ « temps » est le temps entre la réception de l'ordre de départ et le passage du kart.

Extraits des données :
Table TClient

idClient	Nom	Prenom	Niveau	mail
1	Dupont	Paul	3	null
2	Martin	Jeanne	3	null

Table TSerieClient

idSerieClient	noSerie	pseudo	meilleurTour	noKart	idClient
100	40	Polo	*NULL*	2	1
103	40	Jeannette	*NULL*	5	2

Document technique DT8
Partie 1 et 4 – Lampes Philips© Hue

1. Extraits de documents constructeurs

Pont de connexion Philips© Hue

Maximum 50 ampoules ou luminaires par pont.
Protocole Zignee Light link :1.0 certifié.

Installation :
Le pont est connecté au réseau grâce au câble Ethernet fourni.
Les lampes sont automatiquement connectées au pont via le réseau Zigbee.

Lampe Philips© Hue Go

LED 6 W.
16 millions de couleurs.
Lumière blanche fonctionnelle.
Toutes les nuances de blanc, des températures de couleur chaudes à froides.
5 effets dynamiques et 2 effets statiques.

2. Programme permettant d'allumer la lampe numéro 1 grâce aux services REST

Note : REST (Representational State Transfert) est un style d'architecture qui permet, généralement via le protocole HTTP, d'accéder à une ressource définie par son URI (Uniform Ressource Identifier). Différentes opérations peuvent être effectuées telles que la lecture (GET), l'écriture (POST), la modification (PUT), la suppression (DELETE). La représentation des données n'est pas imposée mais les formats JSON et XML sont souvent utilisés.

```
// ordre pour allumer
string MessageBody = "{\"on\":false}";
byte[] messageContent = UTF8Encoding.UTF8.GetBytes(MessageBody);

// transmission de la requête
WebRequest request = WebRequest.Create(new Uri(
"http://192.168.1.17/api/5c1faee2351999fb7a28a8bf1c0c6071/lights/1/state"));
request.Method = "PUT";
request.Timeout = 10000;
request.ContentType = "application/json";
request.ContentLength = messageContent.Length;
Stream stream = request.GetRequestStream();
stream.Write(messageContent, 0, messageContent.Length);
stream.Close();

// lecture de la réponse
WebResponse myWebResponse = request.GetResponse();
Stream ReceiveStream = myWebResponse.GetResponseStream();
```

Document technique DT9

Partie 5 – Le réseau Sigfox©

Sigfox© est un opérateur français de l'IoT (Internet of Things ou Internet des objets en français) qui a développé un réseau bas débit pour connecter des objets à internet.

Le réseau s'étend sur une grande partie de l'Europe, avec environ 2 000 antennes en France. Le très bas débit du réseau (100 bits par secondes) autorise une grande portée (de quelques dizaines de kilomètres en milieu urbain à quelques centaines de kilomètres en milieu rural) et une consommation très faible.
Les objets connectés peuvent envoyer 140 messages par jour, avec une capacité de 12 octets de données utiles dans un message.

Modèle CMEN-DOC v2 ©NEOPTEC

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

Prénom(s) :

Numéro Inscription :

Né(e) le : ☐☐ / ☐☐ / ☐☐☐☐

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

(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.*

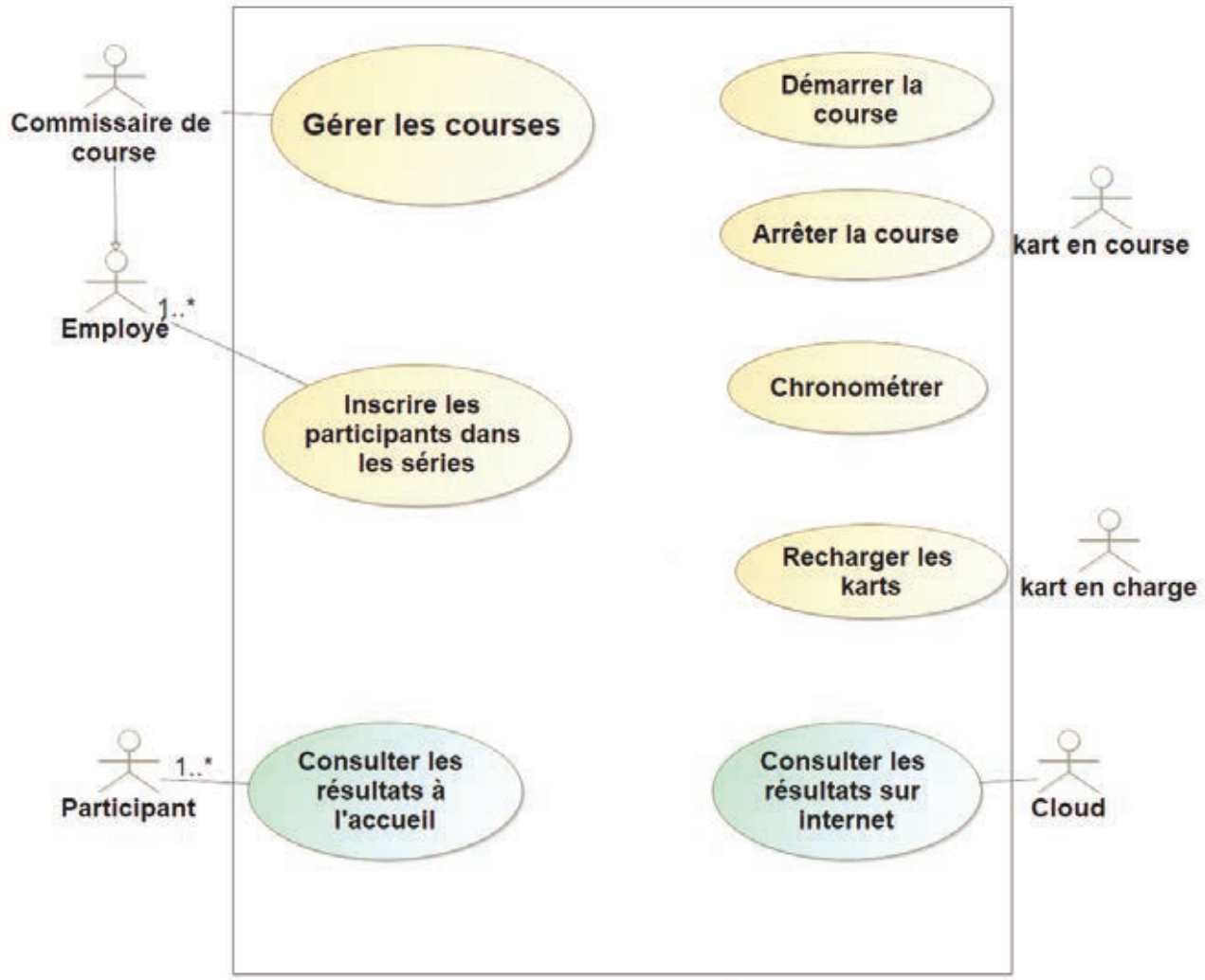
DR1 - DR2 - DR3

NE RIEN ECRIRE DANS CE CADRE

Document réponse DR1

Partie 1 – Analyse préliminaire du système

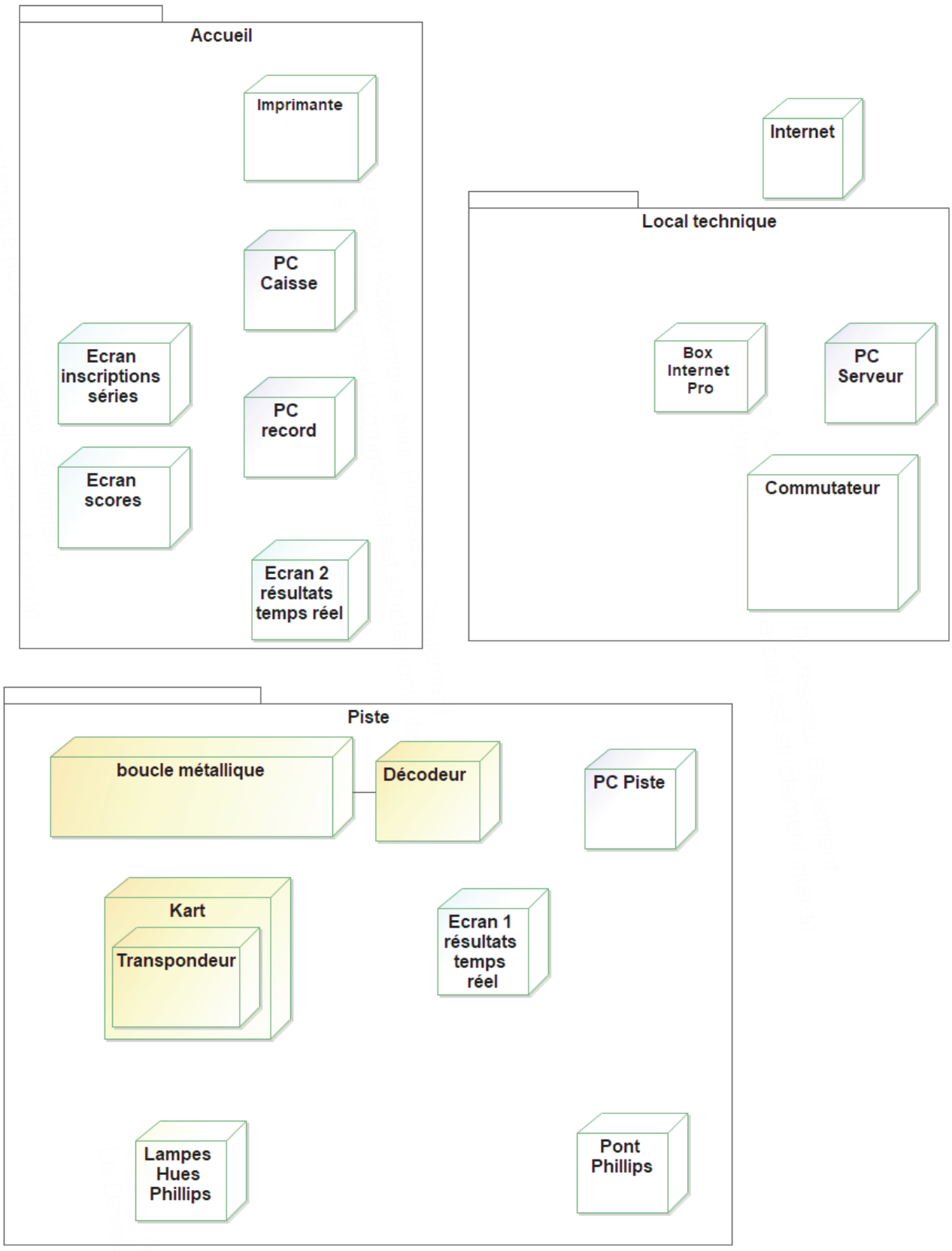
Question 2



Document réponse DR2

Partie 1 – Analyse préliminaire du système

Question 3

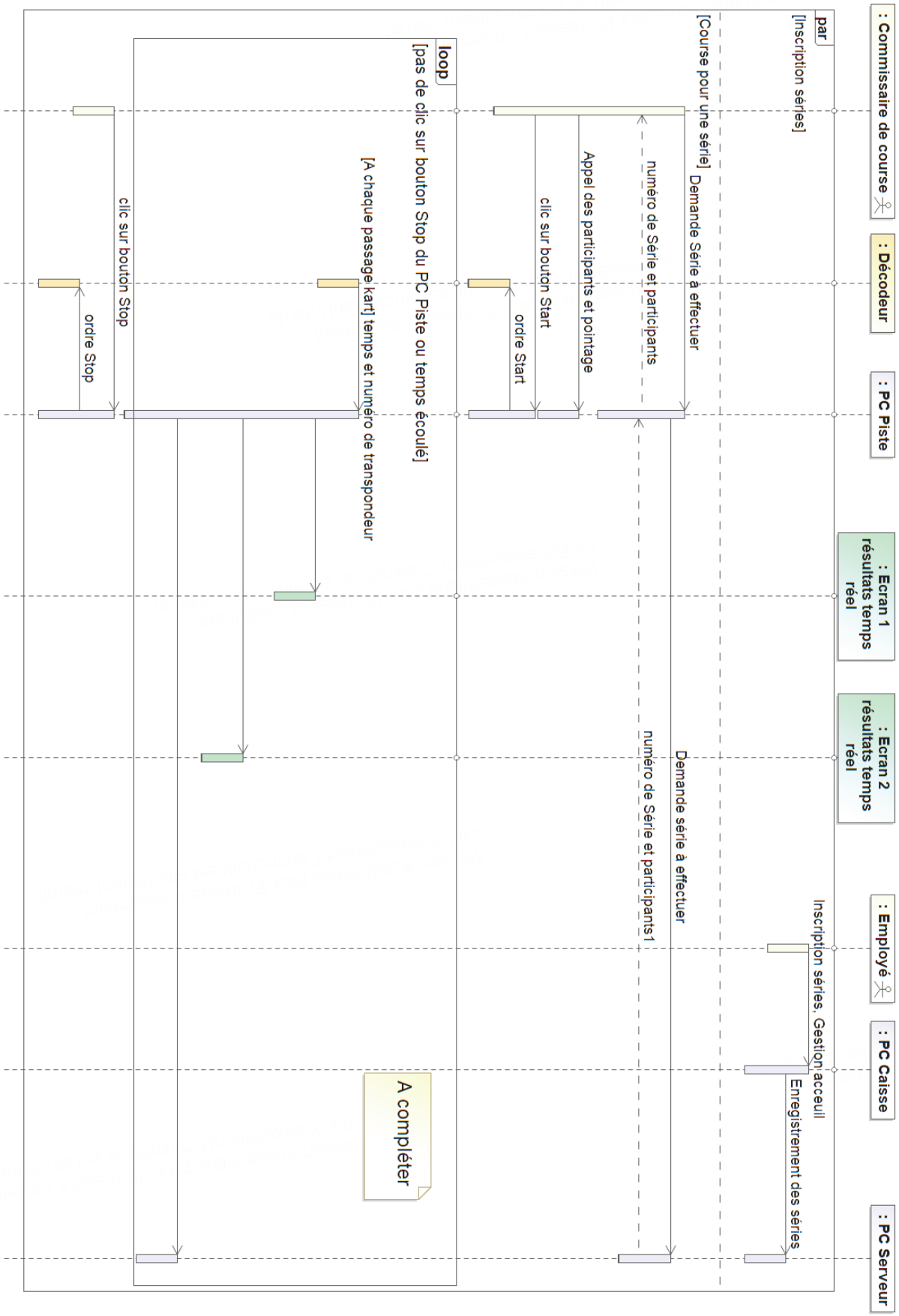


Tournez la page S.V.P.

Document réponse DR3

Partie 1 – Analyse préliminaire du système

Question 4



Modèle CMEN-DOC v2 ©NEOPTEC

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

Prénom(s) :

Numéro Inscription :

Né(e) le : / /

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

(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.*

DR4 - DR5

NE RIEN ECRIRE DANS CE CADRE

Document réponse DR4

Partie 3 – Conception détaillée

Question 23. Compléter les lignes en pointillés.

```
private void serialPort1_DataReceived()
{   // lecture du port série et stockage de la trame dans la variable
    // sTrameLue de type String

   ……..

    if (this.TrameOk(sTrameLue)) // si la trame est correcte
    {
         // si c'est la réponse à un ordre de départ
         if (sTrameLue == "$M;START\r")
         {
             Commun.nEtat = (int)ETAT.MESURE;

         }         // sinon si c'est la réponse à un ordre d'arret
         else if (sTrameLue == "$M;STOP\r")
         {
             Commun.nEtat = (int)ETAT.DEBUT;
         }
         // sinon c'est le passage d'un transpondeur

         else if (……)
         {
                // transmet la trame d'acquittement
                ……..

             // Extrait les informations de la trame
             String sTrameNoKart = sTrameLue.Substring(6, 5);
             String sTrameTemps = sTrameLue.Substring(17, 9);
             String sTrameBatterie = sTrameLue.Substring(15, 1);
             String sTrameSignal = sTrameLue.Substring(12, 2);
             // Enregistre les données dans la file d'attente
             SMesure mesure;
             mesure.no_tour = 0;

            ……..

            ……..

            ……..

            ……..

           ……..
         }
      }
```


Tournez la page S.V.P.

}

Document réponse DR5

Partie 3 – Conception détaillée

Question 25.

idTour	Temps	Signal	Batterie	idSerieClient

Document réponse DR6

Partie 4 – Mise en réseau des équipements

Question 35.

Nom de la couche réseau dans le modèle OSI	Numéro de la couche réseau dans le modèle OSI	Protocole	Principales informations transmises
	7		
	4		
	3		
	2		