



IO-TRUCKS



Présentation Générale du projet	2
Analyse de l'existant	3
Expression du besoin	4
Description structurelle du système	4
Planification	5
Répartition des tâches	6
Partie Etudiant 3 : Mathieu Arthur	7
Objectifs	7
Diagramme des cas d'utilisation	7
Itérations	8
Partie Physique	8
Liaison Bluetooth	8
Ressources Logiciel	9
Android Studio	9
BOUML	9
Subversion	10
Android	11
Diagrammes de classe	12



Présentation Générale du projet

Il s'agit de réaliser un système numérique intégré à un véhicule industriel permettant d'interagir à courte portée sur les équipements intégrés du véhicule tel que le triangle, le hayon, les éclairages de confort et les gyrophares. De plus le système doit être capable de les visualiser en plus de visualiser la charge du véhicule. Ce système est crucial pour l'employé. Il doit être simple, ergonomique et accessible.





Analyse de l'existant

Electro Maintenance est une entreprise spécialisée sur le marché des accessoires et équipements pour véhicules industriels.

Activités : Fabricant et distributeur de pièces détachées électriques, hydrauliques et accessoires destinés aux professionnels de la route.

Ses clients sont : Carrossiers, aménageurs, constructeurs, concessions PL, VUL et Agricole, loueurs, TP, collectivités, etc...

Cette société ne cesse d'innover et d'étendre sa gamme, notamment dans le domaine des objets connectés avec son offre Atlas Connect.

Le système Atlas Connect permettra par exemple :

- de piloter l'éclairage
- de commander l'ouverture/fermeture de porte
- de monter/descendre la benne ou le hayon

Electro Maintenance recherche des solutions pour les besoins suivants :

- triangle rabattable
- détecteur de surcharge ...
- supervision de hayons
- protection contre le vol de marchandises
- éclairage de confort en périphérie du camion





Expression du besoin

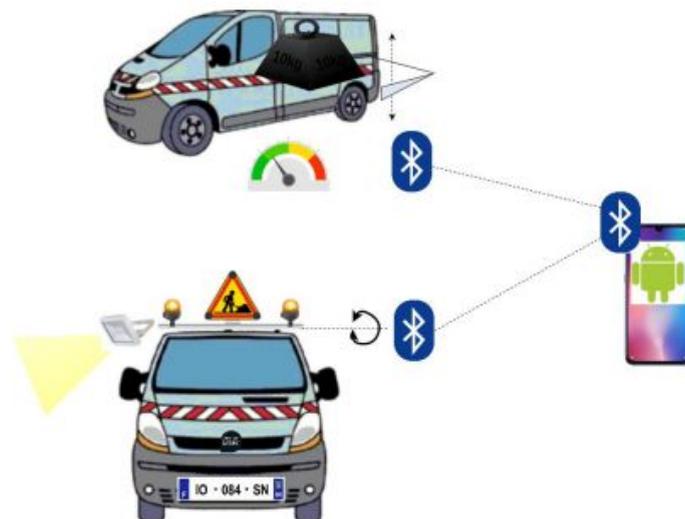
Le système « lo-Trucks » devra remplir les missions suivantes :

- déployer un triangle de signalisation fixé sur le toit du camion
- signaler l'état d'une intervention (par feux de balisage et gyrophare)
- piloter les éclairages périphériques (projecteur en périphérie du camion)
- acquérir les données de fonctionnement (état du triangle, des éclairages, surcharge du véhicule, ...) du camion ;
- assurer la transmission des données des états du camion via une communication sans fil ;
- afficher les données de fonctionnement reçues du camion sur l'écran du terminal mobile ;

Description structurelle du système

Le camion « lo-Trucks » sera équipé :

- d'un système pilotant le triangle de signalisation ;
- d'un système de transmission de données sans fil Bluetooth ;
- d'un système mesurant le niveau de chargement du camion ;
- d'un système de supervision de hayon ;
- d'un système d'éclairage de confort (en option).





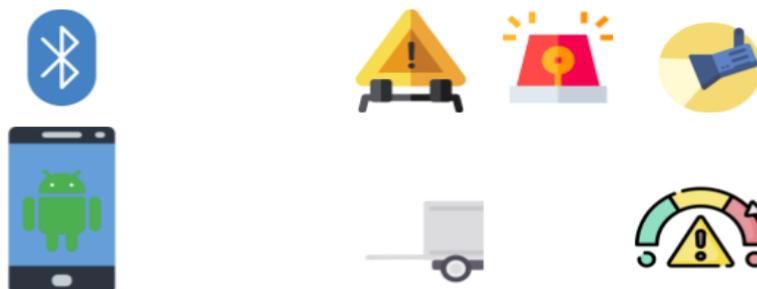
Planification

The screenshot displays a project management application interface with a dark theme. The background is a close-up photograph of the front of a silver Mercedes-Benz truck. The application is organized into five vertical columns representing project phases: 'A Faire EC1 - VASSOUT', 'A Faire EC2 - EL HOUKSSOUS', 'A Faire IR1 - MATHIEU', 'En cours', and 'Terminé'. Each column contains a list of tasks with progress bars and completion dates. The 'En cours' column is the most detailed, showing tasks like 'Communiquer avec le terminal mobile' (May 1), 'Évaluer le niveau de chargement du camion' (1/4), 'Appairer les accessoires' (1/2), 'Contrôler les accessoires' (3/4), and 'Visualiser les informations associées aux accessoires' (3/4). The 'Terminé' column shows completed tasks such as 'Piloter les feux de signalisations du triangle' (Apr 1), 'Piloter les gyrophares (option)' (Apr 1), 'Commander la montée/descente du hayon' (0/4), 'Diagramme d'exigence' (Feb 14), 'Diagramme cas utilisation' (Feb 14), 'Protocole communication', 'Diagramme de bloc' (Feb 14), 'Activité Android Studio' (2/2), 'Maquette IHM', 'Diagramme de classe', and 'Activité Bluetooth'. Each task is accompanied by a progress bar and a checkmark indicating its status.



Répartition des tâches

Répartition des tâches		
Etudiant 1 EC Jordan Vassout	Etudiant 2 EC Mahmoud El Houkssous	Etudiant 3 IR Mathieu Arthur
<ul style="list-style-type: none"> - Commander la montée/descente du triangle - Informer de la position du triangle (haute/basse) - Piloter les feux de signalisations du triangle - Communiquer avec le terminal mobile Recevoir les ordres du terminal mobile - Piloter les gyrophares (option) - Piloter l'éclairage de confort (option) 	<ul style="list-style-type: none"> - Évaluer le niveau de chargement du camion - Avertir d'un dépassement du niveau de chargement (ponctuel et/ou répétitif) - Communiquer avec le terminal mobile - Commander la montée/descente du hayon 	<ul style="list-style-type: none"> - Appairer les accessoires - Contrôler les accessoires (triangle, hayon, éclairage de confort) - Visualiser les informations associées aux accessoires (déploiement du triangle, niveau de chargement, état de l'éclairage de confort)



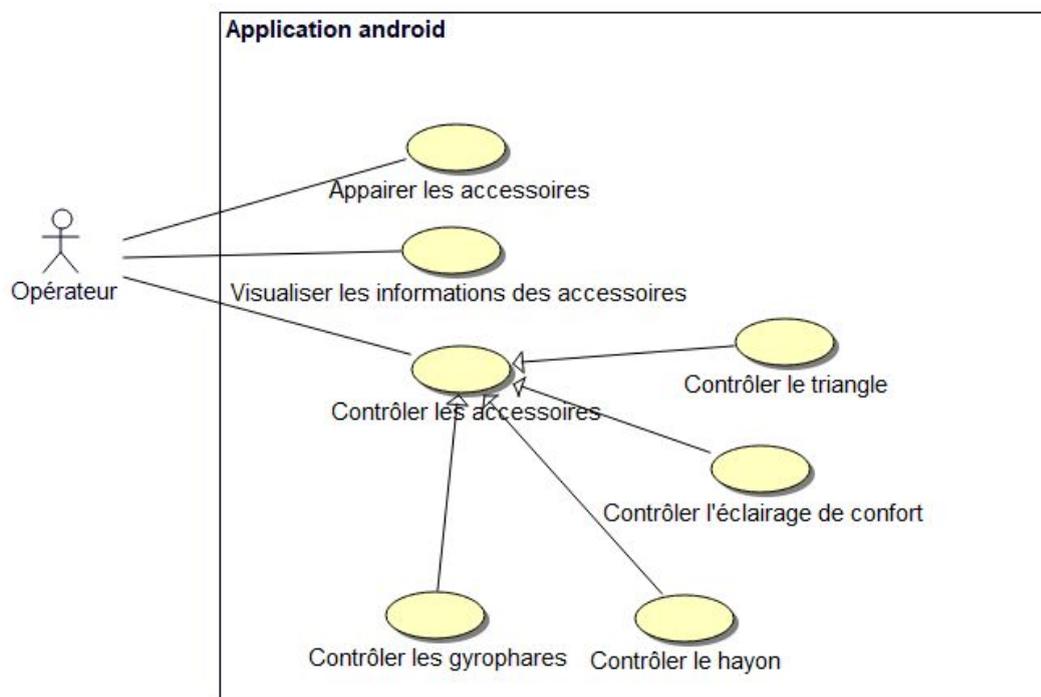


Partie Etudiant 3 : Mathieu Arthur

Objectifs

- Appairer les accessoires
- Contrôler les accessoires (triangle, hayon, éclairage de confort)
- Visualiser les informations associées aux accessoires (déploiement du triangle, niveau de chargement, état de l'éclairage de confort)

Diagramme des cas d'utilisation



L'opérateur a la possibilité d'appairer les accessoires, de visualiser les informations des accessoires et de contrôler les différents accessoires.



Itérations

v0.1	v0.2	v0.3
<ul style="list-style-type: none">- Appareillage bluetooth avec l'esp- Contrôle des gyrophares, éclairages de confort et triangle- Visualisation de l'état des gyrophares, des éclairages de confort et du triangle	<ul style="list-style-type: none">- Contrôle du hayon- Visualisation de l'état du hayon et de la charge- Création du mode Alive	<ul style="list-style-type: none">- Gestion de plusieurs véhicule différents dans une liste- Potentiellement connection à deux cartes différentes

Partie Physique

Liaison Bluetooth



Ressources Logicielles

Android Studio



BOUML





Subversion



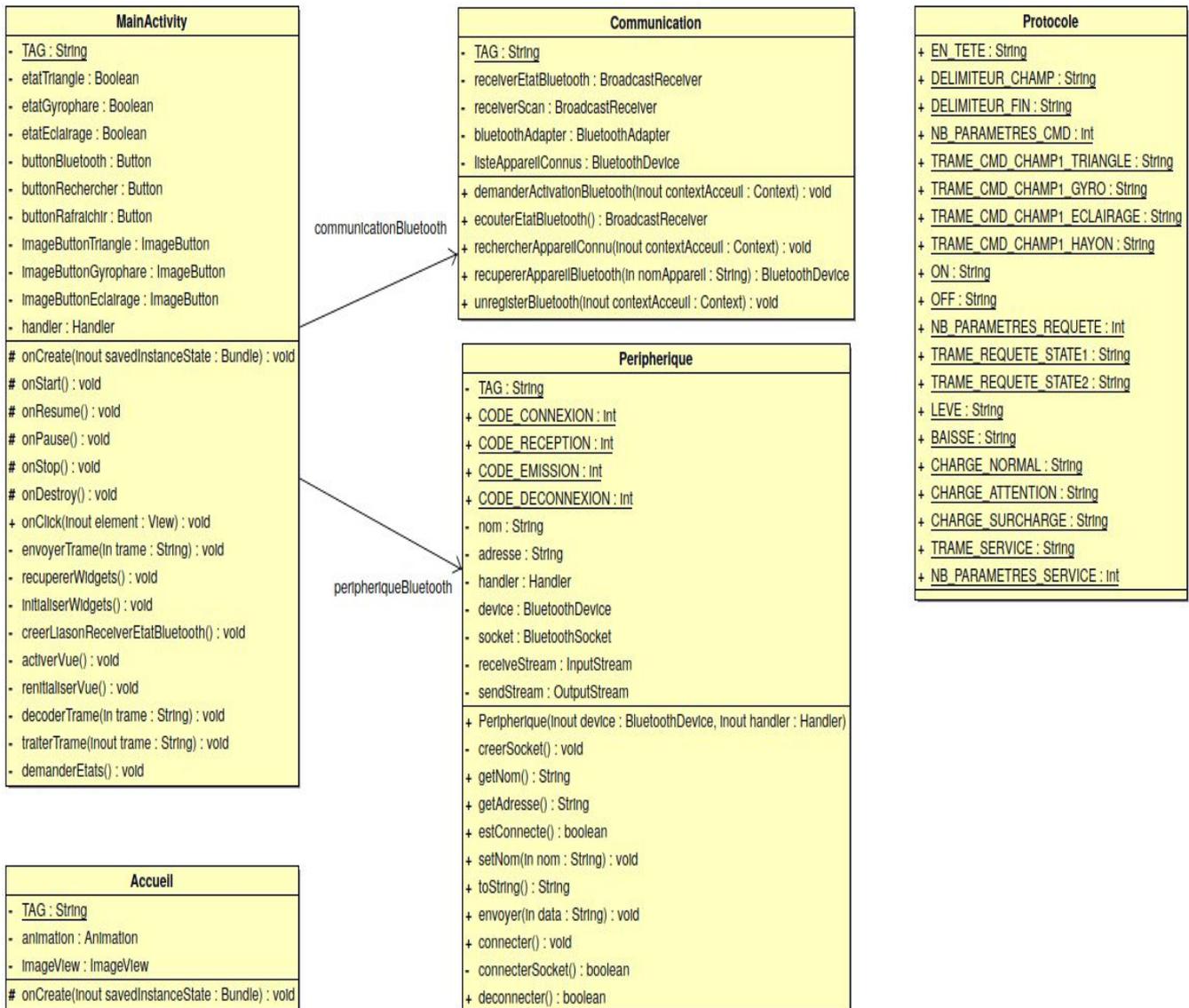


Android





Diagramme de classes



La classe MainActivity représente le menu de gestion et de visualisation des équipements. C'est aussi ici que l'utilisateur cliquera pour se connecter.

La classe Accueil est un *splash* (écran de démarrage fugitif) qui permet de faire une transition.



La classe Communication permet d'établir la communication Bluetooth avec la carte ESP intégrée au camion.

La classe Périphérique est celle qui gère les périphériques Bluetooth auquel on se connecte, et permet d'envoyer les trames.

La classe Protocole permet d'avoir des constantes liées au format du protocole.

La gestion d'application dans Android

Dans le jargon d'Android, une fenêtre d'application est appelée *Activity* ou *Activité*. Une *Activité* contient différents éléments graphiques tels que des boutons, des champs de saisies de texte, des images, etc ...

Chaque *Activity* possède son cycle de vie géré par le système d'exploitation Android. Le système, pour des raisons de priorisation d'activités, peut tuer une activité quand il a besoin de ressources. L'*Activity* peut être dans les différents états suivants :

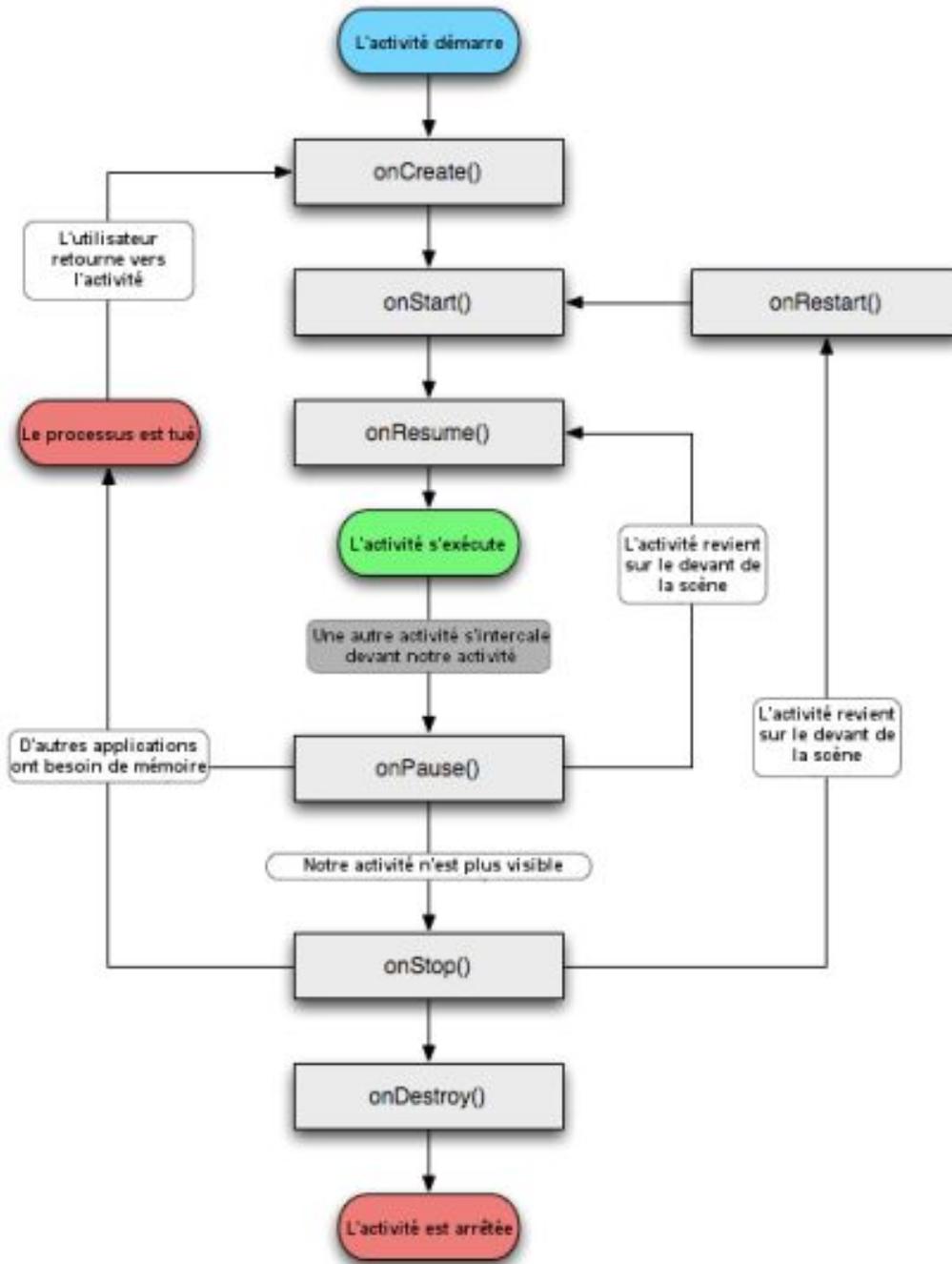
- « Active » : l'activité est lancée par l'utilisateur, elle s'exécute au premier plan
- « En Pause » : l'activité est lancée par l'utilisateur, elle s'exécute et est visible, mais elle n'est plus au premier plan. Une notification ou une autre activité lui a volé la priorité et une partie du premier plan
- « Stoppée » : l'activité a été lancée par l'utilisateur, mais n'est plus au premier plan et est invisible. L'activité ne peut interagir avec l'utilisateur qu'avec une notification.
- « Morte » : l'activité n'est pas lancée.

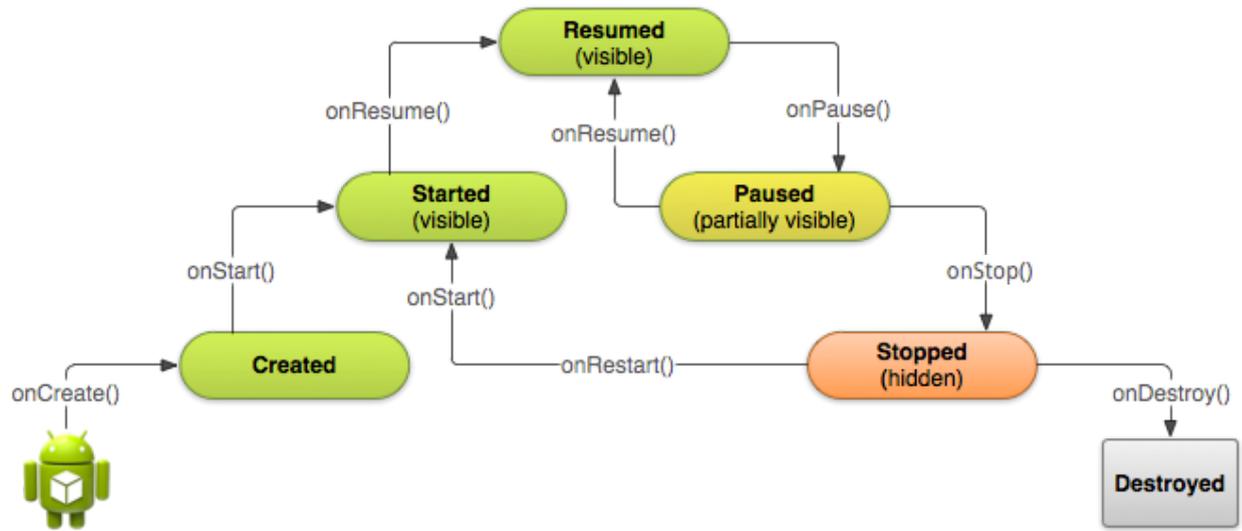
Afin de passer d'un état à un autre l'*Activité* appelle différentes méthodes qui lui sont propres.

- `onCreate()` : elle est appelée lors du premier lancement de l'activité, si l'état du terminal change et que l'activité est associée à cet état par exemple passage du mode portrait au mode paysage. Cette méthode permet aussi les initialisations qui sont effectuées qu'une seule fois au lancement de l'activité.
- `onDestroy()` : est appelé lors de la mort de l'activité qu'elle soit naturelle (créer par l'action de l'utilisateur) ou provoquée par le système par manque de ressources.



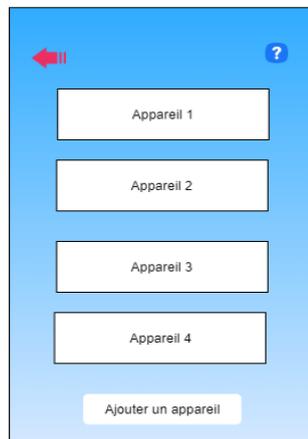
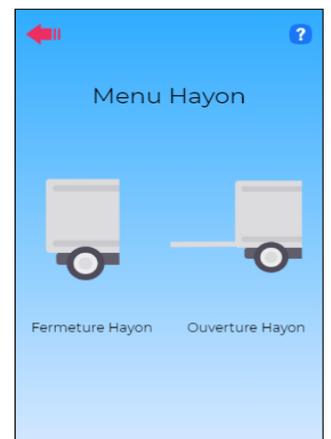
- onPause() et onResume() : sont appelées lors de l'entrée ou la sortie de l'état en pause de l'activité. Elles s'occupent de sauvegarder ses états et les restituer.







Maquette IHM



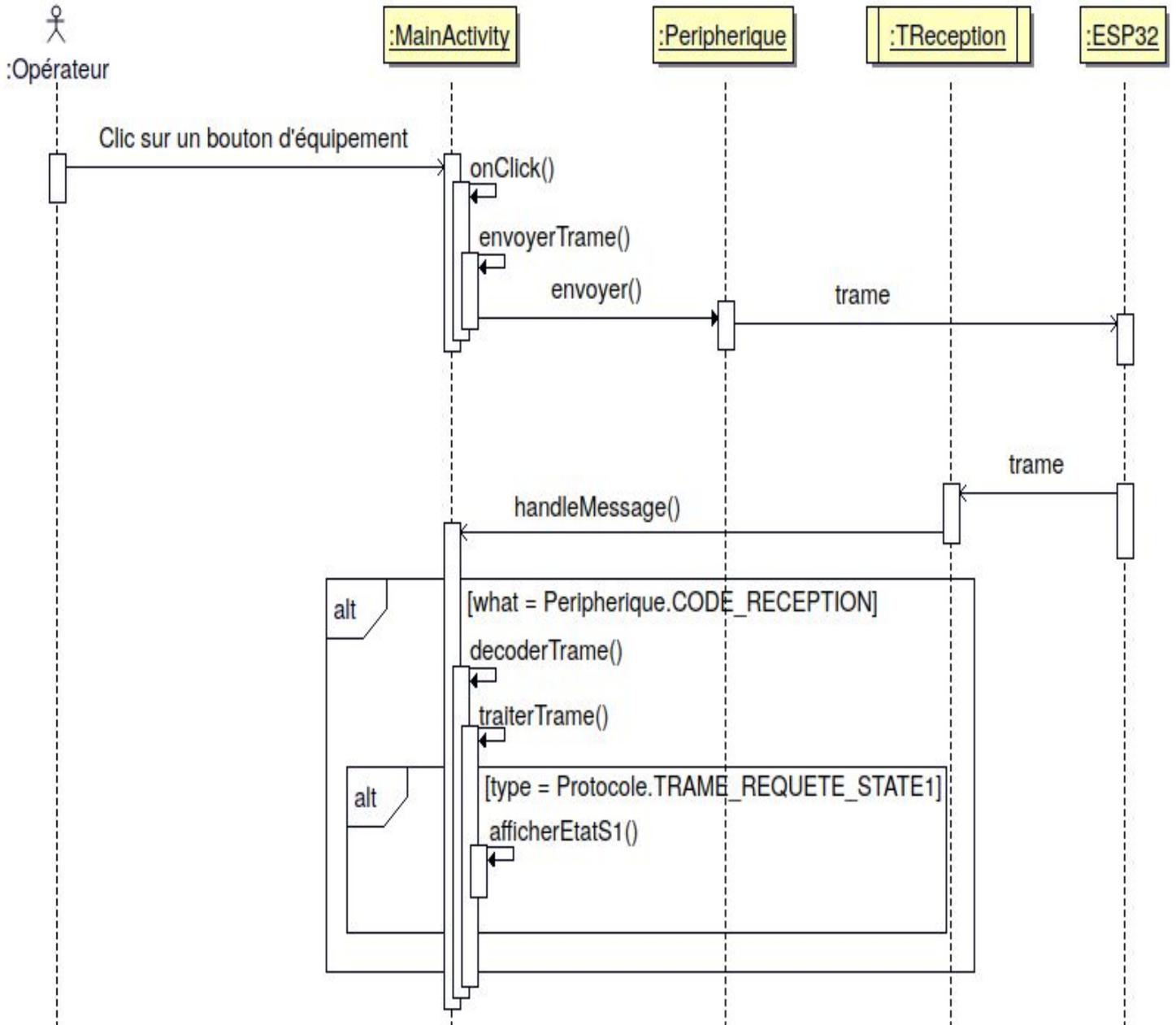


IHM sur la tablette





Scénario





Protocole de communication

Le protocole contient uniquement des caractères ASCII.

Trame de commande (application vers système) :

Format de la trame :

\$iotruck	En-tête
\r\n	Fin de trame
;	Délimiteur
champ 1	le matériel commandé ('T' pour le triangle, 'G' pour les gyrophares, 'E' pour l'éclairage de confort et 'H' pour le hayon)
champ 2	l'état de la commande (dépend du matériel piloté)

Le champ 2 peut prendre les valeurs suivantes :

T (triangle)	1=Lever / 0=Baisser
G (gyrophares)	1=Allumer / 0=Eteindre
E (éclairage de confort)	1=Allumer / 0=Eteindre
H (hayon)	1=Lever / 0=Baisser

L'application envoie une trame pour indiquer au système le matériel qu'elle souhaite piloter et aussi l'état (haut/bas - allumé/éteint)



Décomposition d'une trame

\$iotruck ; **x** ; **x** \r\n

- Début de la trame
- Délimiteur
- Définir le matériel à piloter (T,G,E,H)
- Délimiteur
- Définir son état
- Fin de la trame

Remarque : la trame ne contient aucun espace

Exemples :

\$iotruck;T;1\r\n	→ Lever triangle + allumer leds
\$iotruck;T;0\r\n	→ Baisser triangle + éteindre leds
\$iotruck;G;1\r\n	→ Allumer gyrophares
\$iotruck;G;0\r\n	→ Eteindre gyrophares
\$iotruck;E;1\r\n	→ Allumer éclairage de confort
\$iotruck;E;0\r\n	→ Eteindre éclairage de confort
\$iotruck;H;1\r\n	→ Monter le hayon
\$iotruck;H;0\r\n	→ Descendre le hayon

Trame de requêtes (application vers système) :

Si l'application désire connaître l'état des différents matériels, elle doit émettre une trame de requête.

Partie triangle / gyrophares / éclairage de confort :

Décomposition d'une trame :

\$iotruck ; **S1** \r\n

\$iotruck ; **State 1** \r\n

Remarque : la trame ne contient aucun espace



Partie hayon / niveau de charge :

Décomposition d'une trame :

\$iotruck ; **S2** \r\n

\$iotruck ; State **2** \r\n

Remarque : la trame ne contient aucun espace

Trame d'états (système vers application) :

Les trames d'états sont des réponses aux trames de requêtes émises par l'application.

Partie triangle / gyrophares / éclairage de confort :

Décomposition d'une trame :

\$iotruck ; **S1** ; 1 ; 1 ; 1 \r\n

\$iotruck ; State **1** ; état triangle ; état gyrophares ; état éclairage \r\n

Remarque : la trame ne contient aucun espace

Les différents états pour la trame **S1** :

état triangle	1=Levé / 0=Baissé
état gyrophares	1=Allumé / 0=Eteint
état éclairage	1=Allumé / 0=Eteint

Exemples :

\$iotruck;S1;1;0;0\r\n

\$iotruck;S1;0;1;0\r\n

\$iotruck;S1;0;0;1\r\n

\$iotruck;S1;1;1;0\r\n

\$iotruck;S1;0;1;1\r\n

\$iotruck;S1;1;0;1\r\n

\$iotruck;S1;1;1;1\r\n



Partie hayon / niveau de charge :

Décomposition d'une trame :

\$iotruck ; **S2** ; x ; x \r\n

\$iotruck ; **State 2** ; état hayon ; état charge \r\n

Remarque : la trame ne contient aucun espace

Les différents états pour la trame **S2** :

état hayon	1=Levé / 0=Baissé
état charge	0=Normal / 1=Attention / 2=Surcharge

Exemples :

\$iotruck;S2;0;0\r\n

\$iotruck;S2;1;0\r\n

\$iotruck;S2;0;1\r\n

\$iotruck;S2;1;1\r\n

\$iotruck;S2;0;2\r\n

\$iotruck;S2;1;2\r\n

Trame de service (application vers système) :

L'application envoie périodiquement (toutes les secondes) une trame ALIVE pour maintenir la connexion ouverte.

Décomposition d'une trame :

\$iotruck ; **A** \r\n

\$iotruck ; **Alive** \r\n

Remarque : la trame ne contient aucun espace

Le système répondra par une trame d'acquiescement.

Décomposition d'une trame :

\$iotruck ; **A** \r\n

\$iotruck ; **Ack** \r\n