**Cahier des charges**

# **Robot sauveteur**

**Cahier des charges simplifié :**

1. Réaliser un robot sauveteur
   1. Le robot transmet ce qu’il voit grâce à :
      1. une caméra qui voit dans le visible
      2. une caméra thermique
   2. Le robot a deux modes de déplacements différents
      1. Détection automatique des personnes et approche de la personne
      2. Pilotage manuel à distance via une manette

Répartition des tâches :

RAPHAEL

ALFRED

SALLY

DANIEL

DORIAN

## 

## Partie numérique

### Communication robot - utilisateur

//Raspberry-Raspberry

Ex\_num\_com\_0 : Le robot doit communiquer sans fil avec la partie commande (utilisateur). [0%]

### Partie commande

// Utilisateur-Raspberry

Ex\_num\_cmd\_0 : L’utilisateur doit pouvoir contrôler les mouvements du robot grâce à une télécommande. [90%]

Commentaire : deux cartes raspberry sont à disposition pour le projet

//Ecran-Raspberry

Ex\_num\_cmd\_1 : L’écran doit afficher correctement les images acquises par le robot avec une faible latence.[20%]

Statut : Fonctionne avec l’API sur Windows. Intégration sur Linux en cours

Ex\_num\_robot\_1 : L’utilisateur doit pouvoir connaître la position GPS du robot à tout instant lorsqu’il est à l’extérieur. [90%]

Statut : Le GPS fonctionne mais l’information n’est pas transmise.

## Partie Electronique

Ex\_elec\_robot\_0 : Le robot doit intégrer des éclairages sur sa structure afin d’éclairer son champ de vision jusqu’à de 3-4 m, lorsqu’il se déplace dans des environnements sombres.

→ analyser le courant maximal en sortie de la STM32

→ réaliser 4 modes d’éclairages différents : (éteint, faible,moyen,fort)

Ex\_elec\_robot\_1 : Le robot doit pouvoir se déplacer à une vitesse minimale de 1 km/h

Ex\_elec\_robot\_2 : Le robot doit pouvoir se déplacer avec une autonomie d’environ 30 min [0%]

Statut : théoriquement la batterie devrait le permettre mais le test n’a pas été fait

Ex\_elec\_robot\_3 : Le robot doit pouvoir se déplacer dans des environnements accidentés [0%]

Statut : Les roues ont été conçues de manière à satisfaire cette exigence mais non testé

## Partie Software

### Commande

Ex\_soft\_cmd\_0 : L’écran doit pouvoir afficher les flux vidéos des modules caméras

Commentaire : Les deux flux vidéos ne seront pas affichés en même temps, un bouton “switch” permettra de passer d’une caméra à l’autre. (not a priority)

[10%] : Pour l’instant le flux vidéo est visible uniquement avec un PC windows et il n’est pas transmis à la partie commande

Ex\_soft\_cmd\_1 : La télécommande doit intégrer un bouton pour contrôler l’éclairage

[0%]

Ex\_soft\_cmd\_2 : L’écran doit afficher la position GPS du robot à un endroit non gênant dans l’affichage (not a priority)

[0%]

Ex\_soft\_cmd\_3 : Les commandes issues de la télécommande doivent permettre de piloter le robot

[50%]

Ex\_soft\_cmd\_4 : L’utilisateur doit pouvoir communiquer à distance avec le robot via le module de communication (not a priority)

1. commencer par faire un test de communication entre les deux raspberry

[0%]

### Robot

Ex\_soft\_robot\_0 : Le robot doit envoyer les images de la caméra thermique et de la caméra visible dans un flux organisé. On doit pouvoir recevoir 10 fps au minimum de la part de chaque caméra. (PRIORITY)

Ex\_soft\_robot\_0\_a : Le robot doit pouvoir recevoir le flux d’image et le transmettre dans un flux organisé à un PC de contrôle

[10%] : Pour l’instant le flux vidéo est visible uniquement avec un PC windows et il n’est pas transmis à la partie commande

Ex\_soft\_robot\_1 : Le robot doit envoyer les informations du GPS vers la partie commande toutes les 10 secondes.

[0%]

Ex\_soft\_robot\_2 (optionnel) : Le robot doit envoyer les informations venant des moteurs à la partie commande pour faire de l’odométrie.

[0%]

Ex\_soft\_robot\_3 (optionnel) : Coder le “gouvernail” pour le pilotage de la direction des roues

[90%]

## Partie Mécanique

Ex\_meca\_0 : réalisation du châssis en imprimante 3D

[100%]

Ex\_meca\_0 : réalisation des roues

[100%]

Ex\_meca\_0 : différentiel

[100%]