



Cahier des charges

Ce qu'on nous demande

Poubelles

Hub Arduino

Serveur

Application Mobile

Questions ouvertes

Ce qu'on nous demande

- **3 poubelles avec capteurs / actionneurs**
- **1 HUB Arduino**
- **1 serveur**
- **1 application mobile AR Unity**

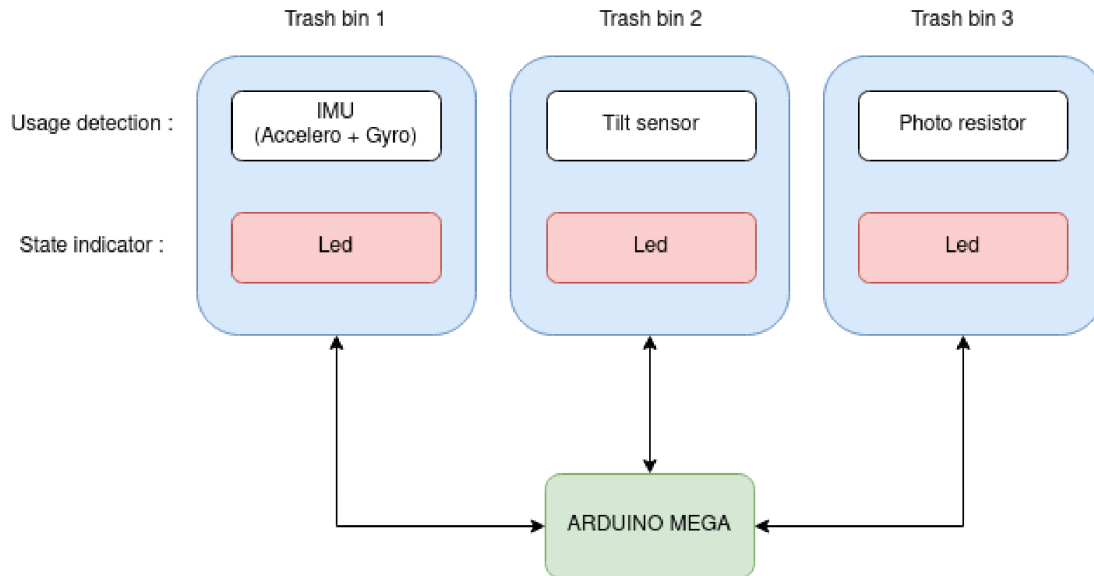
▼ Poubelles

Il faut au moins 3 poubelles contenant des capteurs et actionneurs (leds, moteurs, ...).

Les poubelles seront modélisées à l'échelle ~1:5 (~10cm de large) en impressions 3D.

Ces poubelles seront sous forme de petites impressions 3D

En total: il faut au moins 3 capteurs et 1 actionneur.



Connexion filaire (I²C ?) avec le MCU

Les poubelles doivent être représentées en AR; les poubelles physiques ne sont pas forcément au même endroit que les modèles AR.

Poubelle 0

S'ouvre (avec un moteur) si mouvement détecté (mains / pied ...)

Capteurs :

- **capteur de mouvements / ultrasons** pour ouvrir la poubelle automatiquement

Actuateurs :

- **moteur servo**

Poubelle 1

Capteurs :

- **pavé numérique** pour demander un ramassage
- **capteur de flamme** pour détecter les émeutes

Actuateurs :

- **buzzer(s) passif** pour l'incendie et les sons du pavé numérique

Poubelle 2

S'ouvre avec une carte RFID

Capteurs :

- **Lecteur / Carte RFID** pour verrouiller la poubelle

Actuateurs :

- **moteur pas à pas**
- **LCD** pour afficher divers informations

▼ Hub Arduino

Le MCU est connecté (par fil) au poubelles. Son rôle est d'agréger les données des capteurs et les exposer via une API.

Au démarrage

L'Arduino doit demander l'état des actuateurs pour les mettre à jour.

Communication avec le serveur

La connexion Arduino/Serveur aura besoin de *pings* constants (ex: toutes les 5 secondes), pour permettre de gérer les cas où la connexion est perdue ou le serveur devient inaccessible.

L'Arduino ne permettra qu'une connexion à un client (mobile) en même temps. toute autre tentative de connexion sera rejetée tant que la partie en cours n'est pas terminée.

▼ Serveur

Le serveur est un proxy/broker MQTT Mosquitto (<https://mosquitto.org/>).

Communication avec l'Arduino

Le serveur dicte l'état des actuateurs à l'Arduino.

▼ Architecture suggérée par le Bootstrap

On nous fournit une API HTTP Node-RED qui contient 3 endpoints:

- `GET /status?id=ID&key=API_KEY` retourne l'état actuel du capteur `ID`
- `GET /status_all?key=API_KEY` retourne l'état des **tous** les capteurs.
- `GET /sensors?id=ID&key=API_KEY&...` pour poster un nouvel état pour le capteur `ID`, les données (représentées par `...`) sont sauvegardées dans un fichier `sensor_value_ID`

Dans ce modèle, l'Arduino est un client qui pousse périodiquement l'état des poubelles via `/sensors` et l'appli mobile utilise `/status` et `/status_all`.

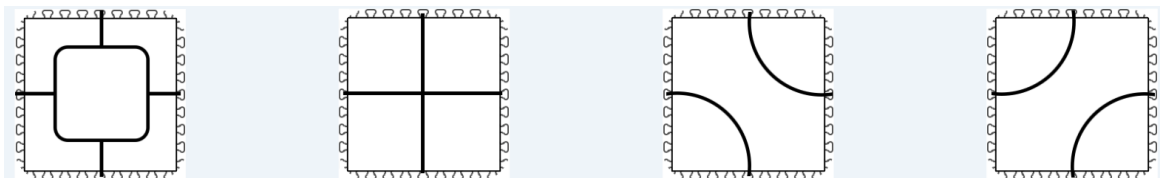
Notre architecture

L'approche du dessus ne semble pas très *scalable*.

Peut être utiliser le protocole MQTT ? (modèle publisher / broker / subscriber)

▼ Application Mobile

Application ARKit/ARCore en Unity qui affiche le trajet d'un camion éboueur sur un trajet défini par des cartes au trajets suivants :



1. Rond-point
2. Carrefour (virages autorisés)
3. Virages
4. Virages 90°

Parcours

La position du camion sera entièrement décidée par l'appli mobile

Le camion commence sa tournée des lors que les 3 poubelles sont plus dans leurs état par défaut.

L'application affiche des représentations des trois poubelles sur le trajet du camion.

Le parcours sera décidé au hasard à la Keynote (parcours **non** pré-déterminé).

Tracking / Cartes AR

On posera des papiers avec des motifs détectables par ARCore, comme des codes QR

(reste à décider le pattern exact).

Peut être possibilité de traquer sans carte ?

Partie Affichage

Les camions et poubelles seront représentées dans un premier temps par des formes simples (des cubes) et plus tard des modèles plus complexes.

Communication avec le serveur

Le client décide seul de la position actuelle du camion.

Interaction avec l'Arduino / Serveur

Quelques suggestions :

- Bouton reset
- Start / stop
- Contrôle de la vitesse
- Contrôle LEDs

Questions ouvertes

- **Quel protocole utiliser pour la communication Arduino ↔ Serveur ↔ Mobile ?**
 - HTTP, HTTPS, ou MQTT?

→ *A priori MQTT, le protocole est supporté nativement par l'ESP 8266*
- **Comment sécuriser l'échange de données entre les parties ?**

- TLS, Chiffrement E2E ?
- L'ESP-8??? semble
- Quelles sont les menaces / vulnérabilités qu'on peut avoir ?
- **Comment faire pour détecter les tapis ?**
 - On détecte les tapis directement ?
 - On colle un code sur papier ?