

Compte Rendu Projet Internet des objets



Lattard
Grégoire
Penne Alexandre
M1 Informatique

Idée : collier gps et suivie d'alimentation
pour chat domestique

Observation du problème :

Plus de la moitié des chats domestiques à travers le monde sont en surpoids, et en France, ce chiffre est d'environ 30%.

Afin de servir des repas quotidiens adaptés au poids, à son mode de vie et à sa castration, le suivi d'une table nous permet d'avoir de savoir comment nourrir nos animaux sans excès. Fractionner le nombre de repas dans la journée aidant à la perte de poids de chats trop gloutons, 5 repas journaliers leurs sont disponibles pour manger à leur faim sans goinfrage.

Un taux qui passe à 50 % chez ceux d'appartement, davantage sédentaires. Permettre un suivi à distance des nos compagnons permettrait de rassurer les plus sceptiques quant à la possibilité de laisser leur chat sortir pour se dégourdir les pattes. Cela permettrait à beaucoup de chats d'appartement de pouvoir sortir.

Objectif : Utiliser le suivi GPS et le calcul d'un régime alimentation adapté

Paramétrage de la palettes node-red et les versions utilisés

node-red v3.1.0

node-red-contrib-web-worldmap v4.5.2 ensemble de noeuds créé par la communauté, permet d'avoir accès à des cartes GPS et avec des interactions api

node-red-dashboard v3.6.1 Pour l'aspect visuel du server

node-red-node-tail v0.4.0 Permet l'utilisation des fichiers locaux

Avant toute utilisation, faites un 'make' à la source du projet.

```
gregoire@gregoire-1-2:~/Documents/Projet_iot$ make
touch Miam.txt
touch GPS.txt
gcc Source/chat_1.c -o chat_1.out
gcc Source/chat_2.c -o chat_2.out
gregoire@gregoire-1-2:~/Documents/Projet_iot$
```

L'activation du serveur de node-red permet à l'utilisateur d'interagir avec l'application, il doit rentrer les informations relatives à ses compagnons.

The screenshot shows a Node-RED interface with two chat forms. The left form features a gauge for 'Croquette restante (g)' at 1000g and a 'REPLIR' button. The right form has input fields for 'Nom', 'Poids', 'Race', and 'Age', along with checkboxes for 'Stérilisé' and 'Sédentaire', and buttons for 'AJOUTER CHAT', 'ANNULER', 'SUBMIT', and 'CANCEL'.

Une fois validé, le récapitulatif des informations de notre chat s'affiche, ainsi que sa localisation.

The screenshot displays two chat detail panels. The left panel for 'Bibi' (Ragdoll, 8 years old) includes a photo, details, and a map. The right panel for 'Chloé' (Siamese, 4 years old) includes a photo, details, and a map. Both panels have a 'SUPPRIMER LE CHAT' button.

Nous imaginons notre collier envoyant régulièrement des données GPS, ainsi qu'un tag RFID associé à l'id du chat ajouté dans la base.

Pour simuler ces envois de données, un logiciel en C nous permet de simuler ces actions pour les chats, nous pouvons le faire manger, ou lui simuler des déplacements.

La simulation se fera à partir de ./chat_1.out et ./chat_2.out

```
choix action Chat 1(rentrez le numero correspondant):
1 - Faire manger le chat
2 - Faire partir le chat a l'aventure
3 - fermer

```

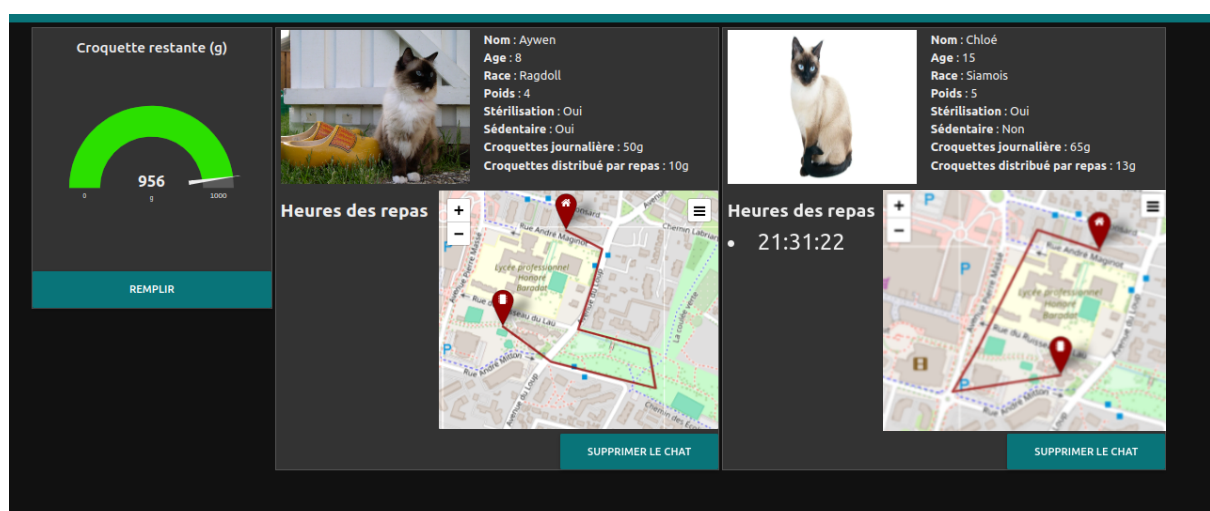
Le choix 1 simule la détection de l'id RFID d'un chat près du distributeur, il envoie un message au topic avec l'id du chat qui veut manger, pour lui servir une de ses doses quotidiennes, on gardera l'heure des repas.

Le choix 2 simule un trajet, une suite de points GPS envoyés par MQTT

```
choix action Chat 2(rentrez le numero correspondant):
1 - Faire manger le chat
2 - Faire partir le chat a l'aventure
3 - fermer
2
2choix du trajet (1 ou 2):

```

2 trajets simulé



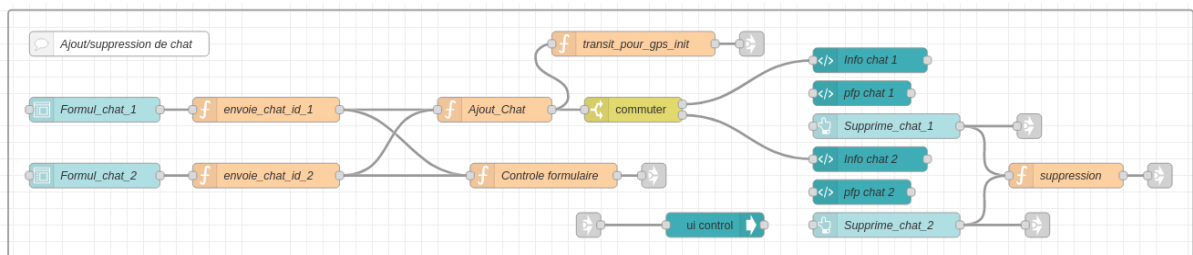
Lors d'un placement de point , la carte se déplace pour suivre son déplacement.

Noeuds et fonctionnement du node-red

Nous avons 3 flow, Server, Distributeur et GPS

Flow Server

- Interaction utilisateur, d'ajout et de suppression d'un chat dans la base



Les formulaire récupèrent les informations transmises par l'utilisateur

Ajout_chat crée une variable globale comportant les informations du chat, c'est à ce moment que son montant quotidien de croquette est défini.

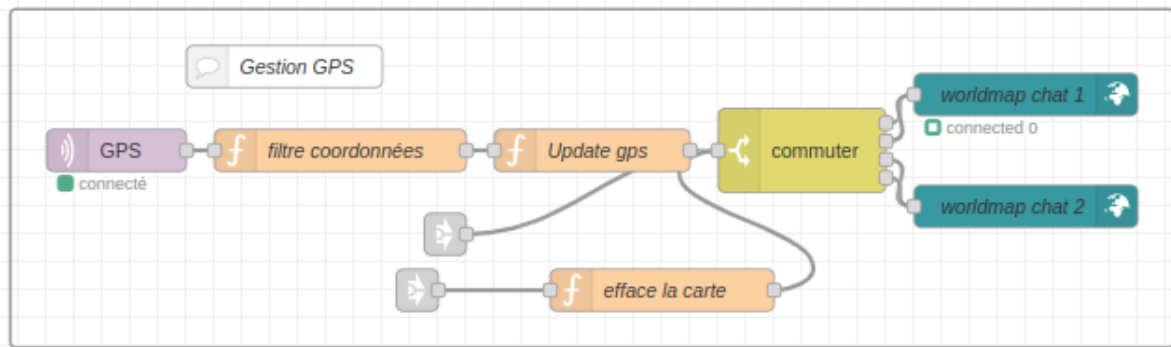
Controle formulaire fait disparaître les formulaires pour faire place au détails du chat et à sa position sur la carte

transition gps init va envoyer au gps d'afficher le point de la maison avec l'icône sur un layer dédié sur les cartes

Quelle quantité de croquettes donner à votre chat en fonction de son poids

Poids du chat	Chat entier et actif	Chat stérilisé OU sédentaire	Chat stérilisé ET sédentaire
2.5kg	45g	40g	30g
3kg	55g	45g	40g
4kg	75g	65g	50g
5kg	90g	80g	65g
6kg	110g	95g	75g

- Gestion visuel de la carte, des trajets



Il faut s'abonner au sujet GPS pour simuler qu'il reçoit en mqtt des coordonnées envoyé par le collier du chat régulièrement (Flow GPS)

Nous recevons des données tel que : "[longitude,latitude]id"

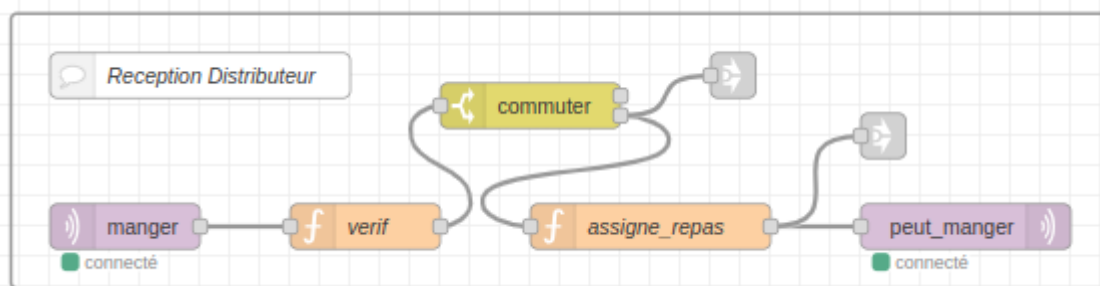
filtre coordonnées sépare en 3 valeurs distinctes pour être traitable.

Update gps récupère la variable global du chat associé à l'id du payload, ajoute dans ses données ses nouvelles positions gps en longitude latitude, et utilise ce tableau de points pour mettre à jours son trajet vers le noeud de la carte **worldmap chat(id)**

Efface la carte survient quand on supprime le chat. On renvoie des coordonnées vide au gps

commute s'occupe d'envoyer les données à la bonne carte, ayant un id de layer par chat.

- Réceptions des id par MQTT pour une demande de croquette



s'abonner au sujet **manger** sur lequel l'id d'un chat est communiqué il simule le scan d'un tag RFID porté par le chat.

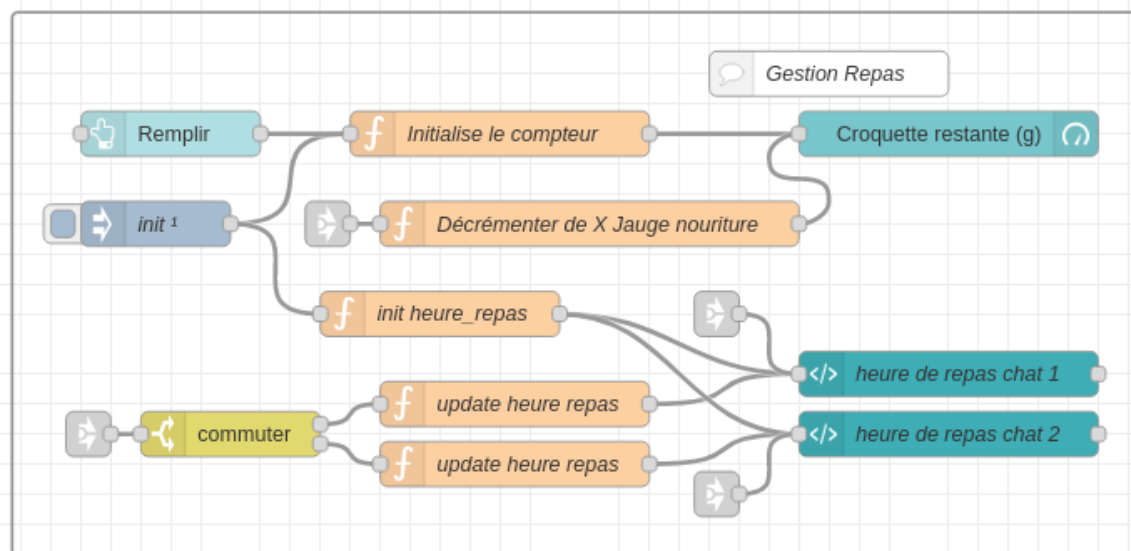
verif regarde si le chat correspondant à l'id reçu n'a pas excéder son nombre de repas quotidiens

commuter traite la vérification, si elle est valable, déclenche l'activation de l'heure vers la gestions des repas

assigne_repas soustrait un repas disponible au chat, transmet son montant de croquette à la gestions des repas, et répond par

MQTT sur le topic `peut_manger` que le chat, peut manger afin que le distributeur puisse s'activer pour donner les croquettes

- Affichage de l'heures des repas, de la jauge de croquette restante



L'**init** initialisé à null les conteneurs html d'**heure de repas** et initialise à 1 kg la quantité disponible du stock de croquettes.

Croquette restante Une jauge qui permet de surveiller la disponibilité en croquette du distributeur.

Remplir simule un remplissage par l'utilisateur du réservoir de croquettes.

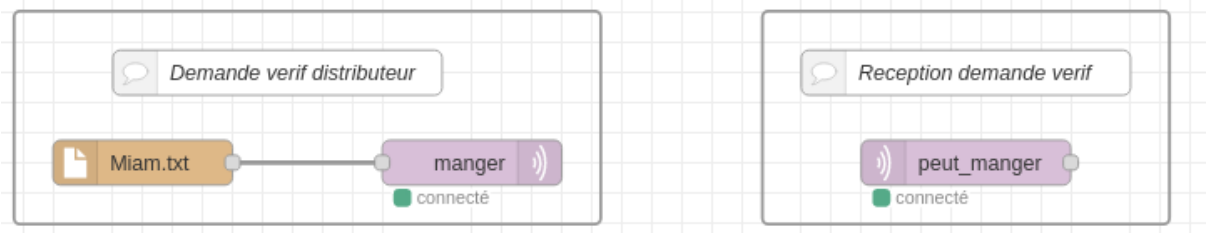
Décrémenter de X jauge nourriture récupère la valeur envoyée par **assigne_repas** (ci-dessus) et enlève le montant du repas à la jauge

commuter récupère l'id du chat en train de manger après la vérification dans la partie précédente et la transmet à la fonction

update heure repas met à jours la liste des heures de repas du chat, transmet un objet avec les heures de repas à l'affichage des **heures de repas**

Flow distributeur

Simuler les interactions des chats



node-tail track si des modifications sont faite au fichier miam.txt modifié par le code C

Comme on simule une simple identification rfid, la donnée écrite sera simplement l'id correspondant au chat

Miam.txt reçoit donc l'id 1 ou 2 du chat souhaitant manger, la demande est envoyée sur le topic manger MQTT.

Après traitement dans le flow serveur,peut_manger permet de signaler au distributeur qu'il peut délivrer les croquettes appropriées

Flow GPS

GPS.txt simule les réceptions de coordonnées GPS, les communique au serveur par MQTT sur le topic GPS, "[longitude,latitude]id" qui passera ensuite dans le filtre gps

