

# Collier analysant le comportement des animaux sauvages

Internet of Things

Annen Rayane, Barreira Romero Adrian, Streckeisen Jarod

11 juin 2024

## Table des matières

<b>Introduction</b>	<b>1</b>
Problématique . . . . .	1
Équipe . . . . .	1
<b>Architecture</b>	<b>2</b>
Matériel . . . . .	3
Conception . . . . .	3
Communication Satellite . . . . .	3
Gestion de l'orientation du collier . . . . .	3
Modèle de classification . . . . .	4
<b>Conclusion et pour aller plus loin</b>	<b>4</b>
<b>Références</b>	<b>5</b>

## Introduction

Ce rapport présente le projet de collier permettant de classifier les comportements d'animaux en fonctions de données récoltées par des capteurs (accéléromètre et gyroscope).

## Problématique

Ce projet fait suite à une proposition de M. Mitouassiyou, un projet similaire a été fait à l'école au sein de l'institut IICT. Toutefois le premier prototype créé par l'institut a encore quelques défauts.

La communication entre le collier et internet se fait par satellite. Le collier doit être donc équipé d'une antenne spéciale, afin de garantir que la transmission des données s'effectue correctement à chaque passe de satellite, cette antenne doit être orientée vers le haut. La première mesure pour qu'elle soit orientée vers le haut a été de mettre un système de contre-poids (voir Figure 1).

Le principal problème de cette conception est que le système de contre-poids est lourd et dérange l'animal qui porte le collier et peut affecter la manière dont il se comporte.

Notre contribution sera donc de fournir un collier qui devra répondre aux critères suivants :

- Collier léger
- Basse consommation d'énergie
- Communication par satellite

## Équipe

- Rayane Annen : communication satellite et intégration des différents composants dans le dispositif

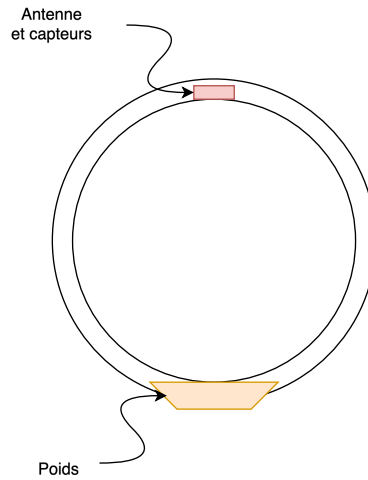


Figure 1: Coupe simplifiée du premier prototype de collier par l'IICT

- Adrian Barreira Romero : modèle de classification du comportement en fonction des données des capteurs obtenus
- Jarod Streckeisen : gestion de l'activation des antennes en fonction de l'orientation du dispositif (et donc du collier).

## Architecture

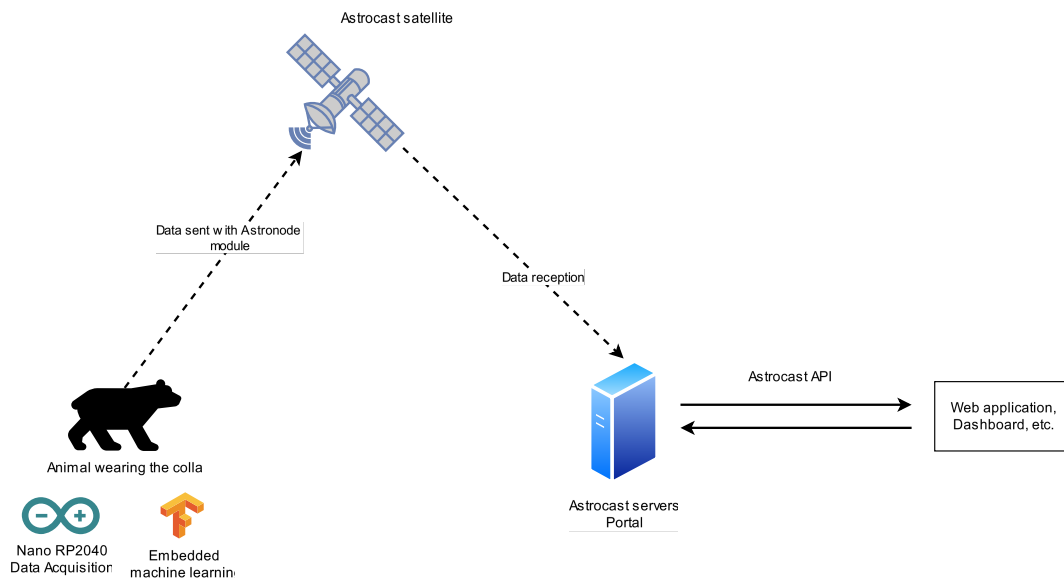


Figure 2: Architecture du système

L'architecture présentée en Figure 2 est celle du système dans son ensemble et comment elle pourrait être imaginée dans un cas futur.

L'animal porte le collier, celui-ci transmet à des intervalles d'environ 30 minutes des nouvelles données vers le portail Astrocast en utilisant leur réseaux de satellites, les données sont calculés à partir d'un modèle directement intégré dans le dispositif RP2040.

## Matériel

Pour réaliser le prototype du projet nous avons eu besoin de :

- Un Arduino Nano RP2040 Connect
- Un developement kit d'Astronode qui embarquait un Astronode S+ et une carte Wi-Fi permettant de simuler le réseau satellite.
- Du matériel électronique de base pour le cablage et le prototypage (résistances 220  $\Omega$ , LEDs, jumper cables)

En vue de finir le prototype, nous avons eu besoin de :

- 4x Antenna Patch Astronode
- Un multiplexeur / demultiplexeur permettant de switcher sur quelle antenne on veut activer.
- Une batterie

## Conception

L'idée de base est la suivante : installer plusieurs antennes (des patches) et programmer un mécanisme qui, en fonction de l'orientation du collier active une ou l'autre antenne. Nous avons ensuite schématisé à quoi devrait ressembler le collier (voir Figure 3)

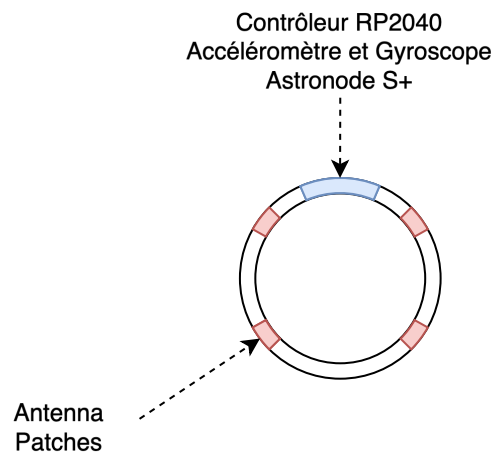


Figure 3: Coupe simplifiée du prototype de collier qu'on a réalisé

## Communication Satellite

Concernant la communication satellite, nous avons utilisé une librairie créée par Astrocast permettant de configurer le module Astronode et l'envoi de message.

Par dessus, on a créé un wrapper permettant d'interagir avec la librairie simplement et de pouvoir tester l'envoi de message sans avoir de module Astronode connecté directement en simulant simplement l'envoi de message.

Les messages sont directement envoyés chez Astrocast et sont accessibles directement sur le portail. Dans un travail futur il est envisageable de pouvoir utiliser leur API pour connecter un dashboard ou tout autre application Web.

## Gestion de l'orientation du collier

Notre première tentative pour définir l'orientation du collier était d'utiliser simplement le gyroscope. Nous nous sommes ensuite rendu compte que le gyroscope ne permettait que d'avoir l'orientation entre 0 et 90 degrés, sauf que nous devons avoir une orientation sur 360 degrés.

Nous avons découvert qu'il était possible de calculer le roulis "roll" en utilisant les quaternions. Malheureusement, malgré les différentes tentatives, nous ne sommes pas arrivés à quelque chose de concluant.

Nous avons donc décidé d'utiliser le filtre de Madgwick. C'est un algorithme de fusion de capteurs (Fusion sensor algorithm). Ces algorithmes sont des techniques mathématiques qui combinent des données provenant de plusieurs capteurs afin de fournir une estimation plus précise et plus fiable de l'état d'un système ou d'un environnement.

Le filtre de Madgwick utilise les données du gyroscope et de l'accéléromètre. Pour ce faire, il utilise la descente de gradient pour optimiser un quaternion qui oriente les données de l'accéléromètre par rapport à une référence connue de la gravité. Ce quaternion est pondéré et intégré au quaternion du gyroscope et à l'orientation précédente. Le résultat est normalisé et converti en angles d'Euler.

Nous avons utilisé l'implémentation arduino officiel ([librairie](#)) de Madgwick qui s'occupe de faire les calculs pour nous et permet de récupérer le roulis, le lacet et le tangage (roll, yaw, pitch).

Les calculs effectués derrière les décors sont complexes et nous nous attarderons pas dessus.

Néanmoins, il est possible de lire le [papier](#) sur ce filtre.

## Modèle de classification

**Modèle** Nous avons découvert qu'il est possible de faire tourner des modèles de classification sur Arduino avec par exemple la librairie [MicroFlow](#).

Il est donc tout à fait possible d'entraîner un modèle et ensuite le faire tourner sur un RP2040. Malheureusement, nous n'avions pas les connaissances nécessaires pour construire un dataset nous-même et nous avons trouvé que très peu de documentation en ligne sur la possibilité de prédire un comportement animal à partir des données d'un gyroscope et d'un accéléromètre.

Toutefois nous avons trouvé un dataset ([Kamminga et al. 2019](#)) provenant d'un prototype similaire qui prédisait le comportement d'un cheval, nous avons utilisé ces données pour créer un modèle de type MLP. Le but final étant de montrer qu'il est possible d'embarquer un modèle dans la puce pour directement effectuer les prédictions.

Dans notre cas, nous avons utilisé les données d'un cheval. Nous avons ensuite entraîné le modèle et réussi à uploader les valeurs sur l'arduino à l'aide de MicroFlow. Par soucis de matériel il a été difficile d'entraîner ce modèle (temps et matériel) le dataset utilisé a donc été très réduit afin de pouvoir produire ce modèle "prototype" dans les temps.

L'upload sur l'arduino fonctionne bien cependant les prédictions sont un peu étranges. Les valeurs peuvent peut-être varier en fonction de la sensibilité des accéléromètres et gyroscopes, mais nous n'avions également pas de cheval ni un collier assemblé à lui passer autour du cou afin de tester. Quand bien même, rien que la constitution du collier ou le cheval en question peuvent faire varier les données d'accéléromètre et gyroscope.

Ainsi, il est très difficile de vérifier que le modèle fonctionne correctement. Mais il est également difficile de prédire le comportement d'animaux du à tous ces éléments.

En bref, les modèles pour prédire le comportement des animaux sont prometteurs mais nous avons besoin de plus de matériel (Données, PCs, etc.) afin de pouvoir réaliser la tâche demandée. Par ailleurs, il est important de souligner que si un modèle doit pouvoir prédire les différents comportements des plusieurs animaux, la demande en données et matériel requis explose encore plus.

Concernant l'intégration du modèle de classification avec les autres composants, celle-ci n'a pas été effectuée par manque de temps.

**Fonctionnement** Tous les documents se trouvent dans le dossier `model`. Le modèle est entraîné dans le jupyter notebook `collar.ipynb`. Ce notebook contient également de quoi générer 2 fichiers, un fichier avec les informations de notre modèle (export pour MicroFlow) et un fichier de code arduino d'exemple (`model_parameters.cpp`)

Les informations du modèle ont été copiées et collées dans le setup de notre code arduino (`arduino_code.ino`).

## Conclusion et pour aller plus loin

Nous sommes arrivés à la conclusion qu'il était tout à fait possible de construire un collier qui respecte les contraintes qui nous ont été données. Nous avons pu avoir un système qui arrive à connaître sa rotation et envoyer des données selon son orientation.

Concernant le travail futur, dans les grandes lignes, il faut intégrer le modèle avec les autres composants, effectuer le montage des composants électroniques et calibrer les capteurs afin d'activer les bonnes antennes selon l'orientation.

## Références

Kamminga, Jacob W., Lara M. Janßen, Nirvana Meratnia, et Paul J. M. Havinga. 2019. « Horsing Around - A Dataset Comprising Horse Movement ». *Data* 4 (4): 131. <https://doi.org/10.3390/DATA4040131>.